

**UNIVERSIDADE FEDERAL DO PARANÁ**

**FABERSON GREIN**

**SISTEMA DE INFORMAÇÃO APLICADO AO CONTROLE DE ESTÁGIOS DO  
CURSO DE GESTÃO DA INFORMAÇÃO DA UNIVERSIDADE FEDERAL DO  
PARANÁ**

**CURITIBA**

**2013**

**FABERSON GREIN**

**SISTEMA DE INFORMAÇÃO APLICADO AO CONTROLE DE ESTÁGIOS DO  
CURSO DE GESTÃO DA INFORMAÇÃO DA UNIVERSIDADE FEDERAL DO  
PARANÁ**

Trabalho de conclusão de curso  
apresentado como critério de aprovação à  
disciplina de Pesquisa em Informação II,  
do curso de Gestão da Informação, do  
Departamento de Ciência e Gestão da  
Informação, Setor de Ciências Sociais  
Aplicadas da Universidade Federal do  
Paraná.

Orientador: Prof. Egon Walter Wildauer,  
Dr.

**CURITIBA**

**2013**

*Aos meus pais,  
Osni e Rosilda, meus irmãos,  
meu primo João Paulo Pinheiro  
e meu professor e mestre Egon W. Wildauer.  
Por todo o apoio e sabedoria a mim dedicados.*

## **AGRADECIMENTOS**

Agradeço a Deus pela vida. Ao professor Egon Walter Wildauer pela disponibilidade e ensinamentos acadêmicos e de vida. E a minha família pelo apoio.

“Para obter algo que você nunca teve, precisa  
fazer algo que nunca fez.”

Chico Xavier

## RESUMO

Descreve projeto de desenvolvimento de um sistema computacional, para auxiliar no controle de estágios e centralização das informações auxiliando a tomada de decisão. O desenvolvimento do sistema computacional foi baseado nas especificações e modelagem de dados criados a partir dos requisitos levantados. Foi utilizada a linguagem *JAVA* em conjunto com a linguagem *FLEX* e o banco de dados *MYSQL*; Visando sistematizar o controle dos estágios e tornar o sistema acessível de qualquer lugar que disponibilize acesso à *internet*, além de organizar as informações tornando possível a extração das mesmas de acordo com as necessidades dos usuários. Antes do desenvolvimento do sistema os dados estavam dispersos, eles foram centralizados em um único banco de dados, facilitando a recuperação das informações, o controle dos estágios e dos estagiários. Com o objetivo de tornar os dados acessíveis a quem é de direito e o processo ágil e seguro.

**Palavras-chave:** Sistema de Informação. Gestão da Informação. Desenvolvimento de Sistemas. Análise de Sistemas.

## **ABSTRACT**

Describes project to develop a computational system to assist in controlling stages and centralization of information to assist in decision making. The development of computational system was based on the specification and modeling of data created from the requirements evaluated. The JAVA language was used in conjunction with the FLEX language, and the MYSQL database; Aiming the systematic control of the Internships and make the system accessible from anywhere that offers Internet access, beyond organize the information making it possible to extract the same according to the needs of those who will make the decisions. Before the system development the data were scattered, they were centralized in a single database, facilitating the retrieval of the information, the control of internships and interns. With the objective of make the data accessible to anyone who has access, making the process more streamlined and secure.

**Keywords:** Information System. Information Management. Systems Development. Systems Analysis.

## LISTA DE FIGURAS

Figura 1 - Recursos do sistema de informação .....	23
Figura 2 - Diagrama de casos de uso. ....	41
Figura 3 – Pacote “Dao” .....	57
Figura 4 – Pacote “Model” .....	58
Figura 5 - Pacote “Validator” .....	59
Figura 6 - Pacote “Service” .....	60
Figura 7 - Diagrama de Sequência "AgenteIntegrador" .....	61
Figura 8 - Diagrama de Sequência "AreaAtuacao" .....	62
Figura 9 - Diagrama de Sequência "Atividade" .....	62
Figura 10 - Diagrama de Sequência "Docente" .....	63
Figura 11 - Diagrama de Sequência "Estudante" .....	63
Figura 12 - Diagrama de Sequência "Login" .....	64
Figura 13 - Diagrama de Sequência "Estagio" .....	65
Figura 14 - Diagrama de Sequência "Organizacao" .....	66
Figura 15 - Diagrama de Sequência "Setor" .....	67
Figura 16 - Diagrama de Sequência "TipoOrganizacao" .....	67
Figura 17 - Diagrama de Atividade do sistema.....	68
Figura 18 - Diagrama de estado "AgenteIntegrador" .....	69
Figura 19 - Diagrama de estado "AreaAtuacao" .....	69
Figura 20 - Diagrama de estado "Atividade" .....	69
Figura 21 - Diagrama de estado "Docente" .....	70
Figura 22 - Diagrama de estado "Estagio" .....	70
Figura 23 - Diagrama de estado "Estudante" .....	70
Figura 24 - Diagrama de estado "Organizacao" .....	71
Figura 25 - Diagrama de estado "Setor" .....	71
Figura 26 - Diagrama de estado "TipoOrganizacao" .....	71
Figura 27 - Diagrama de objeto "Organizacao" .....	72
Figura 28 - Diagrama Entidade – Relacionamento.....	73
Figura 29 - Código <i>JAVA</i> .....	81
Figura 30 - Código <i>FLEX</i> .....	82
Figura 31 - <i>Login</i> .....	83



Figura 32 - Tela inicial .....	84
Figura 33 - Tela de estudante .....	85
Figura 34 - Tela de docente .....	86
Figura 35 - Tela de organização.....	87
Figura 36 - Tela de tipo de organização.....	88
Figura 37 – Tela de setor .....	89
Figura 38 - Tela de área de atuação .....	90
Figura 39 - Tela de agente integrador .....	91
Figura 40 - Tela de atividade.....	92
Figura 41 - Tela de estágio.....	93
Figura 42 - Tela de cadastro de usuário.....	94

## LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais .....	39
Tabela 2 - Requisitos não Funcionais .....	40
Tabela 3 - Especificação da tabela "Estudante" .....	74
Tabela 4 - Especificação da tabela "Docente" .....	75
Tabela 5 - Especificação da tabela "Setor" .....	76
Tabela 6 - Especificação da tabela "Areaatuacao" .....	76
Tabela 7 - Especificação da tabela "Atividade" .....	76
Tabela 8 - Especificação da tabela "Atividadeestagio" .....	77
Tabela 9 - Especificação da tabela "Agenteintegrador" .....	77
Tabela 10 - Especificação da tabela "Organizacao" .....	78
Tabela 11 - Especificação da tabela "Tipoorganizacao" .....	78
Tabela 12- Especificação da tabela "Estagio" .....	79
Tabela 13 - Especificação da tabela "LoginCoe" .....	80

## LISTA DE CASOS DE USO

Caso de Uso 1 - UC01.01 Manter cadastro de estudante .....	43
Caso de Uso 2 - UC02.01 Manter cadastro de organização .....	45
Caso de Uso 3 - UC03.01 Manter cadastro de atividades .....	46
Caso de Uso 4 - UC04.01 Manter cadastro de estágio .....	48
Caso de Uso 5 - UC05.01 Manter cadastro de docente .....	50
Caso de Uso 6 - UC06.01 Manter cadastro de tipo de organização .....	51
Caso de Uso 7 - UC07.01 Manter cadastro de agente integrador.....	52
Caso de Uso 8 - UC08.01 Manter cadastro de setor.....	54
Caso de Uso 9 - UC09.01 Manter cadastro de área de atuação.....	55
Caso de Uso 10 - UC10.01 Manter cadastro de usuários .....	56
Caso de Uso 11 - UC11.01 Logar no sistema .....	56

## LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas  
COE - Comissão de Orientação de Estágio  
DER – Diagrama Entidade Relacionamento  
ER – Entidade Relacionamento  
MER – Modelo Entidade-Relacionamento  
DECIGI – Departamento de Ciência e Gestão da Informação  
DER – Diagrama Entidade-Relacionamento  
SGBD – Sistema Gerenciador de Banco de Dados  
SI – Sistema de Informação  
SQL – *Structured Query Language*  
TI - Tecnologia da Informação  
TIC – Tecnologias da Informação e Comunicação  
UML – *Unified Modeling Language*  
PK – *Primary Key*  
HTML – *Hypertext Markup Language*

## SUMÁRIO

1.	INTRODUÇÃO .....	15
1.1.	CARACTERIZAÇÃO DO AMBIENTE.....	16
1.2.	PROBLEMA .....	17
1.3.	OBJETIVOS .....	18
1.3.1.	OBJETIVO GERAL.....	18
1.3.2.	OBJETIVOS ESPECÍFICOS .....	18
1.4.	JUSTIFICATIVA .....	18
1.5.	ESTRUTURA DO TRABALHO.....	19
2.	FUNDAMENTAÇÃO TEÓRICA.....	20
2.1.	GESTÃO DA INFORMAÇÃO .....	20
2.2.	APRENDIZAGEM ORGANIZACIONAL.....	21
2.3.	INTELIGÊNCIA ORGANIZACIONAL .....	22
2.4.	SISTEMAS DE INFORMAÇÃO.....	22
2.5.	TECNOLOGIA DA INFORMAÇÃO.....	24
2.6.	BANCO DE DADOS.....	25
2.6.1.	DER .....	26
2.6.2.	Normalização.....	27
2.7.	ENGENHARIA DE SOFTWARE .....	28
2.7.1.	UML .....	28
2.7.2.	Ciclo de vida do software.....	29
2.8.	ERGONOMIA DA INFORMAÇÃO.....	30
2.9.	LINGUAGENS DE PROGRAMAÇÃO .....	31
2.9.1.	JAVA.....	31
2.9.2.	HTML.....	31
2.9.3.	FLEX.....	32
2.9.4.	SQL .....	32

2.10.	PADRÃO DE DESENVOLVIMENTO (MVC) .....	33
2.10.1.	MODEL .....	33
2.10.2.	VIEW .....	33
2.10.3.	CONTROLLER .....	33
2.11.	LEI DE ESTÁGIO .....	34
3.	DESENVOLVIMENTO .....	38
3.1.	DESENVOLVIMENTO DO SOFTWARE .....	38
3.1.1.	Descrição do Sistema .....	38
3.1.2.	Escopo do Sistema .....	38
3.1.3.	Requisitos do Sistema .....	39
3.1.4.	Limites do Projeto .....	40
3.1.5.	Especificação .....	41
3.1.5.1.	Diagramas de Casos de Uso .....	41
3.1.5.3.	Diagrama de Classe .....	57
3.1.5.4.	Diagrama de Sequência .....	61
3.1.5.5.	Diagrama de Atividade .....	68
3.1.5.6.	Diagrama de Estado .....	69
3.1.5.7.	Diagrama de Objeto .....	72
3.1.5.8.	Diagrama Entidade-Relacionamento .....	73
3.1.5.9.	Dicionário de Dado e Modelo de Tabelas Normalizado .....	74
3.1.6.	Implementação .....	81
3.1.6.1.	Técnicas e Ferramentas Utilizadas .....	81
3.1.6.2.	Operacionalidade do Sistema .....	83
4.	RESULTADOS E DISCUSSÃO .....	95
5.	CONCLUSÃO .....	96
	APÊNDICE A – CÓDIGO DO SISTEMA PARA “ESTUDANTE” .....	104

## 1. INTRODUÇÃO

O desempenho das organizações na era da informação depende muito se as soluções tecnológicas estão atendendo a demanda da organização, para levar ao cliente a melhor percepção de valor e proporcionar a organização vantagens competitivas, de forma que se destaque no mercado em que atua.

Para melhorar o desempenho da organização são necessárias informações confiáveis e com agilidade, para esse fim são desenvolvidos sistemas computacionais que fazem a coleta e o tratamento dessas informações com maior velocidade e precisão se comparado manualmente.

Com o avanço da Tecnologia da Informação e da Comunicação, as pessoas perceberam que é possível obter, tratar, processar e avaliar informações de forma mais ágil para tomadas de decisões e controle, exigindo cada vez mais dinamismo nos processos informacionais. Além disso, as pessoas esperam que a informação esteja disponibilizada de várias maneiras, diferentes plataformas (celular, notebook, tablets e diversos sistemas operacionais) e que, de qualquer lugar, seja possível fazer o acesso aos mesmos.

Nesse contexto torna-se importante o papel da internet em conjunto com sistemas web para possibilitar o acesso a informações em diferentes lugares e em tempo real, com dados corretamente analisados, tratados e armazenados, possibilitando confiabilidade às consultas.

A ausência de recursos tecnológicos pode prejudicar a oportunidade da organização em melhorar seu desempenho, em atender melhor seu público e gerenciar adequadamente suas informações de modo a prontificar-se em responder às demandas do mercado, às mudanças do ambiente, às dúvidas dos seus interessados, por exemplo: Como fazer uma análise ampla e correta dos dados; benefícios que poderão ser alcançados em termos de redução de custos; como fazer a otimização dos processos, dado que é uma etapa importante para o futuro e o sucesso da organização.

Segundo Oliveira (2009), o perfil dos clientes mudou, eles já nasceram no ambiente digital e buscam nas organizações produtos e serviços para resolver seus problemas de forma ágil, em ambientes digitais. Fazendo com que as organizações procurem aumentar seu poder competitivo, seu crescimento no setor, seus investimentos em tecnologias e na busca de melhores práticas na gestão de seus processos, principalmente no relacionamento com os clientes.

Para fins de estudo deste projeto, o ambiente analisado foi da Comissão de Orientação de Estágio (COE) do curso de Gestão da Informação, onde foram analisadas as formas com que são gerenciadas as informações dos estágios pela COE do DEGICI. Também foram identificadas as necessidades de um sistema que organize e armazene as informações para permitir o controle das atividades desenvolvidas pelos estudantes e o grau de envolvimento dos mesmos.

Este controle sistemático, computacional, poderá permitir um melhor aproveitamento das informações, assim como tentar evitar que estudantes desempenhem atividades que não são da área de Gestão da Informação, evitando equívocos. O sistema também se propõe a realizar análises para auxiliar a tomada de decisão dos coordenadores e docentes do curso.

O produto final deste processo será a apresentação do processo de desenvolvimento um sistema de informação para gerenciar as informações dos estágios, a fim de melhorar a qualidade do trabalho de ensino e a formação dos estudantes do curso de Gestão da Informação da UFPR.

## 1.1. CARACTERIZAÇÃO DO AMBIENTE

A Universidade Federal do Paraná está caracterizada como uma instituição de ensino público com autarquia federal, oferta atualmente 57 cursos de graduação na capital do Paraná. O curso de Gestão da Informação, segundo o Ministério da Educação (2012), teve suas atividades iniciadas no 1ª de Fevereiro de 1999 e, de acordo com a Portaria do MEC número 1.027/2004 – SESu, autorizou seu funcionamento, renovado segundo o Art.35 do Decreto 5.773/06.

Ao iniciar suas atividades o curso (em 1999) pertencia ao Setor de Ciências Humanas, Letras e Artes e, em 2001, foi transferido ao Setor de Ciências Sociais



Aplicadas; possui um conceito cinco de acordo com os resultados do ENADE 2009. O departamento conta com 19 docentes que possuem atuação nas áreas de pesquisa, ensino e extensão.

A estrutura hierárquica do curso de Gestão da Informação pode ser considerada como uma “unidade” dentre todos os cursos ofertados pela UFPR.

Todos os cursos da UFPR possuem duas divisões responsáveis pela administração do curso, sendo caracterizadas pela Coordenação do Curso e Departamento do Curso. A Coordenação do Curso é a “ponte” entre os estudantes e a instituição, ouvindo as necessidades e atendendo-as quando possível, já o Departamento é responsável pelas atividades desenvolvidas pelos docentes e projetos de pesquisa de cada um.

O curso ainda tem convênio com instituições de ensino internacionais como *Universidad Carlos III de Madrid* e *Robert Gordon Aberdeen* da Escócia, que permite o desenvolvimento de pesquisas em conjunto. Além do Centro de Consultoria e Capacitação em Gestão da Informação que, de acordo com informações do sítio do curso: “se ocupa do cumprimento de objetivos pedagógicos e de formação curricular, integrados à prática, em serviços de consultoria prestados por equipes que congregam docentes, profissionais e discentes.”.

## 1.2. PROBLEMA

O problema identificado no Curso de Gestão da Informação da UFPR foi a dificuldade em reunir e analisar as informações dos ambientes de estágios e do controle dos estagiários, além da falta de informação (estruturada e acessível) para a melhoria de tomada de decisões. Desta forma, os problemas identificados foram:

- Informações incompletas;
- Informações desestruturadas;
- Falta de controle;
- Informações não acessíveis.

### 1.3. OBJETIVOS

Abaixo serão descritos o objetivo geral e três específicos que norteiam o projeto de pesquisa.

#### 1.3.1. OBJETIVO GERAL

Desenvolver um sistema de informação para gerenciar as informações dos estágios.

#### 1.3.2. OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral foram definidos três objetivos específicos que visam à solução do problema encontrado.

- a) Informatizar os processos de controle de estágio;
- b) Gerar relatórios para sintetizar as informações do banco de dados;
- c) Disponibilizar informações dos estágios para auxiliar os coordenadores do curso de Gestão da Informação e os integrantes da COE a tomarem decisões.

### 1.4. JUSTIFICATIVA

O controle adotado atualmente para os estágios não permite que se tenham informações relacionadas aos estágios de forma ágil e acessível, inviabilizando uma consulta confiável e em tempo hábil para uma melhor tomada de decisão. E como medida para solucionar esses problemas e atingir os objetivos citados, será desenvolvido um sistema computacional para gerenciar as informações dos estágios.

## 1.5. ESTRUTURA DO TRABALHO

Como forma de apresentar o método para atingir os objetivos supracitados, este trabalho está dividido em cinco capítulos.

O primeiro capítulo apresenta a introdução e justificativa ao tema escolhido, apresentando a necessidade do desenvolvimento de um sistema computacional de forma a atingir os objetivos determinados neste trabalho.

O segundo capítulo apresenta os conceitos que foram usados no projeto.

No terceiro capítulo é apresentado o método de desenvolvimento do sistema computacional proposto.

No quarto capítulo são discutidos os resultados que o presente trabalho proporcionou.

No quinto capítulo apresenta-se a conclusão do desenvolvimento do aplicativo computacional proposto pelo trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentados os conceitos que fundamentam o trabalho, eles são de suma importância para se atingir os objetivos propostos, pois é a partir desses conceitos que o projeto será desenvolvido.

### 2.1. GESTÃO DA INFORMAÇÃO

Segundo Robalo Santos (2008), Gestão são as atividades de planejamento, organização, controle, a fim aperfeiçoar os processos e atingir objetivos pré-estabelecidos. Informação é o conjunto de dados que combinados faz sentido para alguém (ROBALO SANTOS, 2008). E a informação no mercado global é tida como necessidade para a sobrevivência e sucesso das organizações, pois segundo Drucker (1993) a informação vem como base e razão para um novo tipo de gestão. E que num curto prazo haverá a troca do binômio capital/trabalho para informação/conhecimento para o sucesso das organizações.

Segundo Marchiori (2002) a Gestão da Informação (GI) é estudada e aplicada com base em três enfoques, o da Ciência da Informação, da Tecnologia e da Administração. Com isso tem-se o conhecimento de várias áreas e que aplicadas juntas conseguem dar um resultado que aumenta a competitividade, o controle, o planejamento dos bens tangíveis e intangíveis, produtos e serviços da organização.

O resultado final da gestão é dar subsídio para a tomada de decisão baseada em informações que auxiliem na obtenção de vantagens competitivas. E para que a GI seja eficaz é necessário que se estabeleçam políticas para o compartilhamento e uso da informação, levantando requisitos de qualidade, confiabilidade, integridade, entre outras características da informação (BRAGA, 2000).

Para a Aprendizagem Organizacional a Gestão da Informação tem como foco o atendimento das demandas, armazenamento e disseminação das informações importantes para a organização de uma maneira geral. Visando a criação de um ambiente favorável para a obtenção de informação, resolução de problemas,

criação de novos produtos, a transferência de tecnologias e métodos, a visão compartilhada, as boas praticas e a distribuição de informação (MONTEIRO e FALSARELLA, 2007).

A gestão da informação é a forma de trabalhar com a informação desde a identificação até o uso e avaliação, sendo que neste ciclo são necessários conhecimentos de varias áreas, e são três as principais, ciência da informação, tecnologia e administração. A ciência da informação busca entender os processos de busca, coleta, disseminação e uso da informação, já a tecnologia disponibiliza meios para facilitar estes processos e a administração foca na organização, conciliação entre estes dois visando à competitividade e as vantagens advindas da gestão eficaz das informações (MARCHIORI, 2002).

## 2.2. APRENDIZAGEM ORGANIZACIONAL

A aprendizagem organizacional é vista como fonte de resolução de problemas, e fonte de vantagem competitiva nas organizações, pois tem como pressuposto a melhora continua de todos, só se tem aprendizagem quando o coletivo aprende (AUDY apud ARGYRIS, 2005).

Os resultados do Processo de Aprendizagem são novos conhecimentos para a empresa, que é gerado a partir da interação entre o conhecimento tácito e explicito. Este processo de interação segundo Nonaka & Takeushi (2000) partem de quatro formas de conversão do conhecimento: socialização, combinação, internalização e externalização. E o processo de Aprendizagem Organizacional esta ligado ao processo de Gestão do Conhecimento.

O aprendizado organizacional tem como base a memória, para isso são necessários que a organização desenvolva processos para gerenciar a memória organizacional, para reter o máximo das experiências dos seus colaboradores. Tendo isso armazenado de forma adequada e acessível é possível compartilhar os conhecimentos, experiências e modelos mentais das pessoas que compõem a organização, implicando em novas percepções e um comportamento modificado, fazendo com que a organização consiga se adaptar e se antecipar as mudanças

conseguindo com isso vantagens competitivas e sobrevivência no mercado (MORESI, 2001).

## 2.3. INTELIGÊNCIA ORGANIZACIONAL

Inteligência organizacional é o conjunto de processos e ferramentas que permitem as pessoas da organização a tomarem decisões acertadas, com o subsídio de informações de qualidade, esse conjunto pode ser composto por técnicas de *Data Mining*, coleta, tratamento, disseminação e uso de informação, sistemas de informação e tudo o que ajuda aos tomadores de decisão a terem informações confiáveis, atuais e acessíveis (ALVES e FALSARELLA, 2009).

De acordo com Moresi (2001), a inteligência organizacional é uma forma da organização se preparar para as batalhas do mercado, tendo conhecimento de si mesma e do ambiente externo, e para conseguir isto é preciso um sistema que colete e armazene as informações necessárias, podendo organizá-las gerando novos conhecimentos e novas informações que auxiliam na tomada de decisão e colaborem para a capacidade de aprendizado.

É o processo que dá à organização o suporte para tomada de decisões, é um sistema que começa no monitoramento contínuo e sistematizado das informações dos concorrentes, como suas estratégias e novos desenvolvimentos, do mercado interno e externo, novos produtos, informações sobre os usuários, quais são seus desejos e necessidades e se estão sendo atendidos, sobre tecnologias, política, economia e tudo que envolve a organização e pode provocar mudanças, com o objetivo de transformar todas essas informações que estavam espalhadas em informações estratégicas e conhecimento para auxiliar na tomada de decisão e estar preparado para mudança, e se possível sempre as prevendo (MORESI, 2001).

## 2.4. SISTEMAS DE INFORMAÇÃO

Para O'Brien (2004) Sistemas de Informação (SI) são pessoas, *hardware*, *software*, redes de comunicação e recursos de dados que coleta, transforma e

dissemina informações em uma organização. Essas partes que compõe o SI são representados e chamados de Recursos de Sistema de Informação como mostra a figura a seguir:



**Figura 1 - Recursos do sistema de informação**

Fonte: O'Brien (2004, p. 6).

Os sistemas de informação têm um papel fundamental no sucesso das organizações num mercado cada dia mais competitivo. De acordo com Laudon (2007) são seis os objetivos organizacionais que as organizações visam atingir:

- Excelência operacional: conseguir eficiência nas operações para maximizar os lucros e as tecnologias e sistemas de informação são muito importantes para atingir os níveis de eficiência e produtividade nas operações;
- Novos produtos, serviços e modelos de negócios: os sistemas de informação e as tecnologias são os principais meios e ferramentas para inovar, pois é a partir deles que normalmente se modifica, melhora os modelos e se criam novos modelos de negocio, produtos e serviços para aumentar os lucros;
- Relacionamento mais estreito com os clientes e fornecedores: uma forma de fidelizar os cliente e fornecedores é conhecê-los, atende-los de forma personalizada, e saber o que eles precisam e na hora que precisam. E nisso as tecnologias de comunicação e os sistemas de relacionamento com clientes e fornecedores estão auxiliando no estreitamento entre duas ou mais organizações;
- Melhor tomada de decisão: a disponibilização de dados e informações em tempo real com precisão, sem ficar se baseando em intuição, sorte, é uma das

principais aplicações dos sistemas de informação, que dão subsídio para os administradores, na hora certa, com a informação certa para tomar decisões;

- Vantagem competitiva: Se atingido dois ou mais dos objetivos citados, provavelmente vai obter algumas vantagens competitivas, como relacionamento com clientes e fornecedores, novos produtos, serviços e modelos de negócio, melhor tomada de decisão. E se esses processos fizerem com que a organização consiga produtos e serviços superiores com custos mais baixos, ela aumentará os lucros, e os concorrentes não conseguirão chegar ao seu nível de qualidade, preço, etc;
- Sobrevivência: os sistemas de informação estão sendo cada vez mais indispensáveis nos negócios sejam eles de pequeno, médio ou grande porte, pois possibilita o acompanhamento das mudanças muitas vezes impostas pelo mercado.

Segundo Rezende (2008), são diversos os conceitos sobre sistemas, mas ele cita alguns que mais se destacam:

“[...] conjunto de partes que interagem entre si, interagindo-se para atingir um objetivo ou resultado; partes interagentes que formam um todo unitário com determinados objetivos e efetuam determinadas funções; em informática, é o conjunto de *software*, *hardware* e recursos humanos; componentes da tecnologia da informação e seus recursos integrados; empresa ou organização e seus vários subsistemas.”

E segundo o mesmo autor, sistema de informação é todo o sistema que consegue manipular dados e que gera informações, usando ou não recursos computacionais. Podendo assumir diversos formatos como relatórios, procedimentos, fluxos, ou qualquer documento que tenha sentido para alguém, e que facilite a gestão da organização.

## 2.5. TECNOLOGIA DA INFORMAÇÃO

De acordo com Rezende (2008), a Tecnologia da Informação (TI) é o meio usado para receber, processar e disseminar informações, ela é necessária para que os SI possam estar acessíveis a todos os envolvidos nos processos que ele



esta envolvido. São os hardwares e softwares que auxiliam no desenvolvimento e aplicação dos sistemas. Segundo Padoveze (2002) a “Tecnologia da Informação é todo o conjunto tecnológico à disposição das empresas para efetivar seu subsistema de informação”.

São recursos necessários para a guarda, recuperação, uso, geração, controle e coleta de informações, sendo eles o *hardware*, *software*, sistemas de telecomunicações e gestão de dados e informações, mas esses recursos não fazem nada sozinhos, precisam ser manuseados, organizados e sistematizados por pessoas para que desempenhem seus s papeis (REZENDE, 2008).

É indiscutível o papel da Tecnologia da Informação nas organizações, num mercado cada vez mais dinâmico, informações precisando ser transferidas *on-line*, necessidade de tomada de decisão baseada em informações e conhecimentos acessíveis a qualquer hora, e quem não se adequar a nova realidade e a nova forma de tratar as informações, possivelmente não terá sucesso no mercado. E a TI auxilia os negócios a serem mais flexíveis, administrar as informações, a apoiar a comunicação, disseminação e uso do conhecimento, com recursos computacionais como *software*, *hardware* e redes de comunicação (LEHMKUHL et al, 2008).

## 2.6. BANCO DE DADOS

Os bancos de dados facilitaram os processos de armazenagem e acesso a dados, através o seu conjunto de softwares chamado “sistema gerenciador de banco de dados (SGBD)”, pois eles permitem acessos simultâneos, evitam problemas de duplicação e incoerência e permite um controle de segurança, integridade que era difícil de conseguir. Eles são projetados para trabalhar com outras aplicações a fim de proporcionar facilidade na coleta, recuperação, uso e disseminação, esse projeto de banco de dados começa na etapa de modelagem de dados, que são diagramas para representar o que será armazenado (TURBAN, 2007).

Para Date (2003), “Um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros”, e segundo o mesmo autor

o banco de dados (BD) pode ser comparado com prateleiras que estocam livro, jornais, só que no caso foi projetado para armazenar dados eletrônicos que tenham significado para a organização a quem o sistema irá ser desenvolvido e é capaz de desempenhar funções de inserção, deleção, alteração, busca e remoção, o sistema de banco de dados é composto por *hardware*, *software*, usuários e dados.

KORTH e SILBERSCHATZ (1989) discorrem que:

*“Um sistema de gerenciamento de banco de dados (data-base management system - DBMS) consiste numa coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. A coleção de dados é comumente referenciada como o banco de dados, que contem informação de um particular empreendimento. O principal objetivo de um DBMS é proporcionar um ambiente, conveniente e eficiente, para retirar e armazenar informação no banco de dados.”.*

#### 2.6.1. DER

O diagrama entidade-relacionamento (DER) tem o objetivo de facilitar o projeto de banco de dados a partir da abstração do mundo real onde as entidades são objetos como “carro” e “estudante” e os relacionamentos mostram as ligações entre esses objetos, permitindo que o sistema seja especificado graficamente (KORTH e SILBERSCHATZ, 1989).

O DER é uma notação para a modelagem de dados, a principal função é a de representar os objetos de dados e suas relações de forma gráfica. Um DER é composto por entidades, que vão se tornar pelo menos uma tabela e relacionamentos entre elas, REZENDE (2005).

Segundo COSTA (2006) a abordagem ER foi criada em 1976 por Peter Chen. Essa técnica é a mais utilizada para modelagem de banco de dados e é composta por três conceitos:

- As entidades, que representam objetos do mundo real, abstratos ou concretos, como, pessoas, produtos, contas.

- Os atributos, que descrevem os objetos, ou seja, as entidades, por exemplo, a entidade pessoa é composta pelos atributos, nome, CPF, idade, RG.
- E os relacionamentos, que representam ligações entre as entidades. Por exemplo, a entidade “pessoa” esta ligada a entidade conta.

Para TURBAN [et al.] (2010), as entidades são representadas por retângulos, as elipses representam os atributos e os quadrados representam os relacionamentos.

### 2.6.2. Normalização

Para Date (2003), normalização de banco de dados é o processo que visa uma melhor definição das tabelas de um banco de dados, evitando falhas na construção do banco como, redundância e dados diferentes na mesma tabela. Uma técnica muito usual é a utilização das três formas normais. Uma entidade esta bem definida quando esta na Terceira Forma Normal (3FN), mas para chegar nela precisa passar pela Primeira Forma Normal (1FN) e Segunda Forma Normal (2FN). Abaixo serão apresentados, de acordo com Date (2003), os conceitos de cada forma normal:

- 1FN – todos os atributos de uma entidade devem ser atômicos, ou seja, tem que corresponder a um único valor.
- 2FN – precisa estar na 1FN e todos os atributos que não são chave dependam somente da chave primária.
- 3FN – precisa estar na 2FN e não conter dependências transitivas entre os atributos, eles devem ser mutuamente independentes.

O processo de normalização segundo Heuser (2004) se baseia no conceito de forma normal, que são regras que devem ser seguidas pelas entidades para que estas estejam bem projetas e possam ser usadas sem ocorrência de problemas como redundância e multiplicidade de dados. As formas normais são denominadas como simplesmente primeira, segunda, terceira e quarta forma normal, abreviadamente 1FN, 2FN, 3FN e 4FN.

## 2.7. ENGENHARIA DE SOFTWARE

Com o surgimento do *software*, e o seu uso cada vez mais indispensável e intenso, o número de novos sistemas foi muito grande, só que a maioria desses sistemas desenvolvidos por volta de 1970 não era entregue ou eram entregues com falhas, nasceu então à crise do *software*. E para tentar aumentar a qualidade e produtividade desses sistemas foi criada a disciplina de Engenharia de Software para padronizar e desenvolver métodos que garantissem a eficácia do software (PRESSMAN, 2006).

Segundo Rezende (2005), Engenharia de Software é uma metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas: adequação aos requisitos funcionais do negocio do cliente e seus respectivos procedimentos pertinentes: efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da Informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas.

Os objetivos da Engenharia de Software segunda MAFFEO (1992), são o aprimoramento da qualidade dos produtos de software e o aumento da produtividade dos engenheiros de software, além do atendimento aos requisitos de eficácia e eficiência.

### 2.7.1. UML

Para Guedes (2006), “A UML (Unified Modeling Language ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos”.

A UML é uma linguagem para a especificação de sistemas computacionais, que teve origem em 1996 a partir de três metodologias, são elas, método de Booch, método *Object Modeling Technique*, de Ivar Jacobson e o método Object-Oriented Software Engineering de James Rumbaugh. Essa junção foi financiada pela Rational Software. Em pouco tempo as empresas adotaram essa nova metodologia e em 1997 a UML foi adotada como linguagem padrão para

modelagem pela OMG (Object Management Group ou Grupo de Gerenciamento de Objetos) (GUEDES, 2006).

A UML é uma **linguagem visual** para modelar sistemas orientados a objetos. Isso quer dizer que a UML é uma linguagem constituída de elementos gráficos (visuais) utilizados na modelagem que permitem representar os conceitos do paradigma orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema. (MENEZES, 2002).

### 2.7.2. Ciclo de vida do software

Uma grande parte dos produtos são criados, usados e descartados, e com o software não é diferente, ele passa pelas fases de concepção, a partir da identificação de uma necessidade, é desenvolvido, nas fases de análise e programação, e entra na fase de operação, a partir disso ele pode ser descartado a media que não atende mais as novas necessidades, ou melhorado e ampliado para atender os novos processos (PAULA FILHO, 2003).

Segundo Rezende (2005) quando um software não sofre modificações (manutenção, novas implementações) o tempo de vida dele é de no máximo cinco anos. Por isso têm-se o conceito de que não existe um software pronto, pois com as mudanças de processo e utilização do sistema, são realizadas manutenções, correções e melhorias. Para Rezende (2005), são basicamente oito as fases de vida de um software, sendo elas:

- Concepção – idealização do software;
- Construção – análise e programação;
- Implantação – testes e disponibilização para os usuários;
- Implementações – ajustes depois da implantação;
- Maturidade e utilização plena – Uso em produção, com o sistema atendendo as demandas para qual foi idealizado;
- Declínio – quando o sistema não atende a todas as necessidades que precisa;
- Manutenção – correções e adaptações para atender as novas necessidades. Nesta etapa o software fica num ciclo de melhorias e aumento das funcionalidades;
- Morte – descontinuidade do sistema.

## 2.8. ERGONOMIA DA INFORMAÇÃO

De acordo com Montmollin (1990), ergonomia pode ser denominada como a ciência do trabalho, porem muito autores dão definições diferentes e o que torna a área bem abrangente, mas confusa. O sentido etimológico do termo ergonomia é o governo da força, e tem como foco o estudo das pessoas em seus postos de trabalho levando em consideração os aspectos físicos, psíquicos e sociais.

A ergonomia teve início com a 2ª Guerra Mundial entre 1939 e 1945, onde os países começaram a usar a força da tecnologia como aliada, só que com essa inovação vieram os problemas com a alta exigência dos operadores com operações complicadas, altas cargas de tensão, alta pressão ambiental, física e psíquica. Foi quando o engenheiro K. F. Hywell Murrel, o fisiologista Floyd e o psicólogo Welford estudaram formas de adaptar os aparatos bélicos a fim de diminuir as perdas e acidentes com os soldados e materiais (MORAES, 2005).

Como fica fácil de perceber estes estudos não focaram só as características físicas da pessoa, foram abordados também aspectos psicológicos e cognitivos como a recepção e tratamento de informações (ABRAHÃO & PINHO, 2002). E segundo Moraes (2005) esse grupo de profissionais formou em 1949 a “*Ergonomic Research Society*” a fim de estudar o homem no posto de trabalho, visando às máquinas que as pessoas usam no dia a dia para trabalhar e verificaram várias falhas nos processos, como postura, locais inadequados, falta de segurança e muitas outras que exigiam muito dos trabalhadores e com o tempo provocavam lesões. E só a partir dos anos 80 os estudos sobre ergonomia começou a prestar atenção nas cargas informacionais que decorrem do grande número de sistemas e da expansão da internet no ambiente de trabalho.

## 2.9. LINGUAGENS DE PROGRAMAÇÃO

Linguagem de programação é o conjunto de instruções que o homem cria para que a máquina (computador) execute uma tarefa determinada (SAID, 2007).

### 2.9.1. JAVA

A linguagem de programação *JAVA* foi criada por um grupo da *Sun Microsystems* em 1991, com o objetivo de ser usado em dispositivos de consumidor de forma simples e neutra em relação à arquitetura, podendo ser usada em diferentes tipos de *hardwares*. Só que segundo Horstmann (2004), “nunca foi encontrado um cliente para essa tecnologia”.

Em 1994 o grupo da *Sun Microsystems* percebeu que o que tinha sido desenvolvido poderia se tornar um navegador com uma arquitetura neutra, segura, confiável e em tempo real. Podendo ser executado em uma grande variedade de *hardwares*. Ao navegador foi dado o nome de *HotJava* em 1995 na *SunWorld*, onde poderiam ser feitos *downloads* e executar mini aplicativos e programas da *Web*. Esses novos programas denominados *applets* eram escritos na nova linguagem Java e permitiam a criação de várias animações e interação das páginas *Web*. (HORSTMANN, 2004)

### 2.9.2. HTML

Segundo William (2012) o *HTML* teve sua raízes criada em 1980 no projeto baseado em hipertexto com o nome de *ENQUIRE* e em 1989 esse mesmo grupo conseguiu implementar a comunicação entre um cliente *HTTP* e um servidor através da internet criando a *World Wide Web*. Em 1990 surgiu a linguagem *HTML* que significa *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto) e é aplicada para estruturação de documentos e que diferentemente da linguagem anterior *SGML* permite o *link* entre páginas web. No início quem controlava as mudanças era o *CERN* e a *IETF*, após 1995 a linguagem passou a ser padronizada pela *W3C*, organização que regula os padrões *Web*.

Uma segunda versão da linguagem, segundo William (2012) , foi o *HTML 2.0*, que corrigiu algumas falhas da versão anterior, ou seja, não houve mudança na sua estrutura. Em 1999 foi lançada a versão 4.01 do *HTML*, que tinha como função a compatibilidade com as versões anteriores, foram incluídas, de acordo com o mesmo autor, três novas implementações:

***Script*** (estrita) – que proíbi a utilização de elementos obsoletos da linguagem;

***Transitional*** (transitória) – permite elementos obsoletos;

***Frameset*** – que é direcionada para páginas que usam frames.

O *HTML 4.0* mudou as páginas web permitindo melhorias nas interfaces, acessibilidade para pessoas com incapacidades, mecanismos de *scripts* e *style sheets* (CSS) (WILLIAM, 2012).

### 2.9.3. FLEX

A linguagem *FLEX* foi lançada em 2004 pela *Macromedia*, sua função é ser usada em aplicações *Web* com a plataforma *Macromedia Flash*. Em 2005 a *Macromedia* foi adquirida pelo *Adobe*, mudando algumas formas de como as licenças eram comercializadas. A linguagem usa *MXML* e *XML*, para a criação de interfaces, *ActionScript* para facilitar a interatividade com o usuário e linguagem *Flash* (VIRGILIO, 2009).

A estrutura usada é a de Cliente-Servidor, e uma diferença com o *HTML* é que para mudar uma interface não é necessário carregar uma nova página, pois as visualizações se concentram em uma única página, deixando mais rápido. Mas para conseguir rodar a uma aplicação em *FLEX* é necessário que o cliente tenha instalado o *Flash Player* no *browser* (VIRGILIO, 2009).

### 2.9.4. SQL

Segundo Farias (2011), a linguagem *SQL* surgiu em 1970, foi desenvolvido junto a *IBM* para viabilizar o modelo relacional desenvolvido por Edgar Frank Codd, matemático britânico. *SQL* significa *Structured Query Language*, ou Linguagem de Consulta Estruturada.



A linguagem *SQL* é dividida em dois grupos, Linguagem de Definição de Dados (*DDL*) e Linguagem de Manipulação de Dados (*DML*). A *DDL* agrupa os comandos para definir e modificar tabelas do bando de dados, e a *DML* é responsável por manipular os dados de uma tabela com comandos para inserir, excluir, alterar e consultar os dados (FARIAS, 2011).

## 2.10. PADRÃO DE DESENVOLVIMENTO (MVC)

O *MVC* é um padrão de projeto (*Design Patternes*) que utiliza uma arquitetura em três camadas (*3-tier*), onde divide a aplicação para facilitar a divisão das tarefas e da lógica de negócio (PIETRO, 2001).

### 2.10.1. *MODEL*

Responsável por agrupar as características dos objetos que serão manipulados na aplicação. Esses objetos na aplicação são representados pelas classes, como por exemplo, estudante, banco, empresa (PIETRO, 2001).

### 2.10.2. *VIEW*

A visualização é responsável por fazer a interface com o usuário, é a camada que apresenta e captura os para serem usados na aplicação e manipulados pelo controlador (PIETRO, 2001).

### 2.10.3. *CONTROLLER*

É responsável pelas regras de negócio do aplicativo, é a camada intermediária onde os dados são manipulados e consolidados, por exemplo, no banco de dados. É o controlador que irá definir o comportamento da aplicação, nesta camada são definidos os fluxos dos processos de negócio e o comportamento da interface com relação ao usuário (PIETRO, 2001).

## 2.11. LEI DE ESTÁGIO

Na UFPR, o Estágio é regido pela Lei Federal n.º 11.788, de 25 de setembro de 2008, que dispõe sobre o estágio de estudantes, e pela Resolução 46/10 - CEPE /UFPR, que dispõe sobre os estágios na UFPR. Para desenvolver um sistema que controla os estágios na instituição, fazem parte da Normalização Interna da Universidade Federal do Paraná referente a estágios a:

**Resolução n.º 70/04** – CEPE – Dispõe sobre as Atividades Formativas na flexibilização dos currículos dos cursos de graduação e de ensino profissionalizante da UFPR (publicada em 13/07/2004).

**Instrução Normativa 01/03** – CEPE – Normatiza os estágios não obrigatórios (voluntários) previstos na Res. 19/90 – CEPE. (publicada em 14/01/2004).

**Instrução Normativa 01/93** – CEPE/IN – Normatiza os estágios previstos na Res.19/90-CEPE, realizados no exterior.

**Instrução Normativa 01/92** – CEPE/IN – Normatiza os estágios previstos na Res.19/90-CEPE, realizados no âmbito da UFPR.

**Resolução n.º 46/10** – CEPE – Dispõe sobre os estágios na UFPR

Os principais pontos da Lei Federal n.º 11.788, de 25 de setembro de 2008, e da Resolução 46/10 – CEPE /UFPR, são:

1. O Estágio deve ter, fundamentalmente, caráter pedagógico, para a concordância da respectiva Coordenação do curso;
2. No caso de a empresa decidir gerir seu programa de estágio, será formalizado um Termo de Convênio, firmado diretamente com a UFPR, sem ônus para a empresa;
3. Caso a empresa opte por utilizar os serviços de um agente de integração, esse agente também deverá ter um Termo de Convênio com a UFPR. Nesse caso, a empresa concedente do estágio deve pagar ao agente de integração à custa que lhe forem atribuídas;
4. Formalizado o Termo de Convênio, a empresa poderá proceder à seleção e admissão do estagiário conforme a legislação vigente;
5. A contratação do estagiário deve ser formalizada através do Termo de Compromisso de Estágio firmado entre o estagiário e a empresa e na UFPR, com

intervenção do coordenador do curso ao qual pertence o estagiário e da Coordenação-Geral de Estágios;

6. A modalidade do Estágio deve ser curricular ou extracurricular;
7. O Estágio só poderá ser exercido por alunos com matrícula e frequência regular no curso de graduação de Gestão da Informação;
8. O Estágio deverá ter acompanhamento efetivo do professor orientador e do supervisor orientador da parte concedente;
9. Exigir do educando e da empresa concedente a apresentação periódica, em prazo não superior a 6 (seis) meses, de relatório das atividades desenvolvidas;
10. A empresa concedente deverá, por ocasião do desligamento do estagiário, entregar o termo de realização do estágio, com a indicação resumida das atividades desenvolvidas;
11. A instituição de ensino deverá providenciar o arquivamento dos documentos de estágio, como o contrato, relatório e termos de estágios;
12. O horário de estágio será definido de comum acordo entre a instituição de ensino, a parte concedente e o aluno. No caso de estudante do ensino superior, será de no máximo 6 (seis) horas diárias e 30 (trinta) horas semanais, devendo constar o Termo de Compromisso de Estágio;
13. A duração do Estágio, na mesma parte concedente, não poderá exceder a 2 (dois) anos, exceto quando se tratar de estagiário portador de deficiência;
14. A parte concedente fornecerá benefício relacionado a transporte;
15. A eventual concessão de outros benefícios, como alimentação e saúde, não implicam vínculo empregatício;
16. Contratação por parte da concedente, em favor do estagiário, de Seguro contra Acidentes Pessoais, os dados sobre o seguro devem constar no Termo de Compromisso de Estágio;
17. É assegurado ao estagiário receber Bolsa-Auxílio ou outra forma de contraprestação; Também é assegurado ao estagiário – sempre que o estágio tenha duração igual ou superior a 1 (um) ano – um período de recesso de 30 (trinta) dias, a ser gozado preferencialmente durante suas férias escolares;
18. O recesso deverá ser remunerado;

19. Os dias de recesso e a remuneração do estagiário deverão ser concedidos de maneira proporcional, nos casos do estágio ter duração inferior a 1 (um) ano;
20. No caso de prorrogação do Estágio na empresa, deverá ser providenciado o Termo de Aditivo de Estágio no prazo de 7 (sete) a 15 (quinze) dias do término do Estágio;
21. Deverão constar dados sobre conta bancária do estagiário para o recebimento de bolsa;
22. No Termo de Compromisso de Estágio devem constar as assinaturas do coordenador do curso e do representante da unidade concedente de estágio antes de enviar à Coordenação-Geral de Estágios;
23. No Termo de Compromisso de Estágio, devem constar dados relativos à unidade concedente do Estágio;
24. No Termo de Compromisso de Estágio, também devem constar os dados da instituição de ensino;
25. No Termo de Compromisso de Estágio, devem constar ainda as seguintes assinaturas: do estagiário, da unidade concedente, da Coordenação do Curso, da Comissão Orientadora de Estágio, da Coordenação-Geral de Estágio e do agente de integração;
26. O Termo de Compromisso de Estágio deve ser impresso em 4 (quatro) vias, frente e verso;
27. No Termo de Convênio de Estágio, devem constar dados da instituição de ensino e do curso e também dados da empresa concedente.

Diante do contexto acima, foram retirados os elementos essenciais à construção lógica do Banco de Dados para o Controle de Estágio do DECIGI/UFPR.

Na UFPR, o Estágio é regido pela Lei Federal n.º 11.788, de 25 de setembro de 2008, que dispõe sobre o estágio de estudantes, e pela Resolução 46/10 - CEPE /UFPR, que dispõe sobre os estágios na UFPR. Com a intenção de responder à necessidade de se criar um Banco de Dados (BD) que controle os estágios na instituição, torna-se pertinente averiguar como se constitui a Normalização Interna da Universidade Federal do Paraná com referência a Estágio:

Resolução n.º 70/04 – CEPE – Dispõe sobre as Atividades Formativas na flexibilização dos currículos dos cursos de graduação e de ensino profissionalizante da UFPR (publicada em 13/07/2004).

Instrução Normativa 01/03 – CEPE – Normatiza os estágios não obrigatórios (voluntários) previstos na Res. 19/90 – CEPE. (publicada em 14/01/2004).

Instrução Normativa 01/93 – CEPE/IN – Normatiza os estágios previstos na Res. 19/90-CEPE, realizados no exterior.

Instrução Normativa 01/92 – CEPE/IN – Normatiza os estágios previstos na Res. 19/90-CEPE, realizados no âmbito da UFPR.

Resolução n.º 46/10 – CEPE – Dispõe sobre os estágios na UFPR.

### 3. DESENVOLVIMENTO

Neste capítulo serão apresentadas as etapas do desenvolvimento do software e os tópicos que descrevem o sistema computacional, seus requisitos identificados, bem como a especificação e implementação.

#### 3.1. DESENVOLVIMENTO DO SOFTWARE

O processo de desenvolvimento do software consiste num conjunto de etapas para a construção de um sistema informatizado, essas etapas são: descrição do sistema, definição do escopo, análise de requisitos, especificação dos requisitos e implementação.

##### 3.1.1. Descrição do Sistema

O Sistema de Gestão de Estágios tem a finalidade de gerenciar as informações sobre os estágios. Essas informações são:

- Quais as principais atividades desenvolvidas;
- Horários dos estágios;
- Qual o ramo de atividade dessas organizações;
- Quem é o aluno que vai estagiar;
- Quem é a organização onde o estágio será realizado.

##### 3.1.2. Escopo do Sistema

Será desenvolvido um sistema de gestão de estágios para o DECIGI, para gerenciar as informações dos estágios. O sistema computacional irá gerenciar as informações relacionadas aos estagiários, organizações, ramo de atividade das organizações e atividades desenvolvidas pelos estudantes.

Essas informações podem ser utilizadas para a impressão de relatórios, onde poderão ser tomadas decisões para reestruturação do currículo, divulgação do curso e controle dos estágios.

O sistema irá contar com as seguintes funções: cadastrar, alterar e excluir: estudante, organização, área de atuação, setor, tipo de organização, atividades, estágio, agente integrador e docente.

### 3.1.3. Requisitos do Sistema

Analizando os processos referentes aos estágios, foram definidos os requisitos que deverão ser atendidos pelo sistema computacional. A tabela 1 mostra os requisitos funcionais.

REQUISITOS FUNCIONAIS	CASO DE USO
RF01: O sistema deverá permitir manter o cadastro de estudante.	UC01.01
RF02: O sistema deverá permitir manter o cadastro de organização.	UC02.01
RF03: O sistema deverá permitir manter o cadastro de atividades.	UC03.01
RF04: O sistema deverá permitir manter o cadastro de estágio.	UC04.01
RF05: O sistema deverá permitir manter o cadastro de docente.	UC05.01
RF06: O sistema deverá permitir manter o cadastro de tipo de organização.	UC06.01
RF07: O sistema deverá permitir manter o cadastro de agente integrador.	UC07.01
RF08: O sistema deverá permitir manter o cadastro de setor das organizações.	UC08.01
RF09: O sistema deverá permitir manter o cadastro de área de atuação das organizações.	UC09.01
RF10: O sistema deverá permitir manter o cadastro de usuários.	UC10.01

**Tabela 1 - Requisitos Funcionais**

Fonte: Autor

A tabela 2 apresenta os requisitos não funcionais.

REQUISITOS NÃO FUNCIONAIS
RNF01: O sistema deverá ser desenvolvido com a linguagem <i>JAVA</i> e interface em <i>FLEX</i> .
RNF02: O sistema deverá ser executado no browser <i>Chrome</i> .
RNF03: O sistema deverá usar o bando de dados <i>MYSQL 5.5</i> .
RNF04: O sistema deverá acessar o servidor através dos protocolos TCP/IP.

**Tabela 2 - Requisitos não Funcionais**

Fonte: Autor

#### 3.1.4. Limites do Projeto

O projeto do Sistema de Gestão de Estágios irá abordar apenas os itens declarados no tópico 3.1.3, na tabela 1 referente aos requisitos funcionais. O controle de recursos humanos, recursos financeiros, contabilidade, não fará parte do escopo, excluindo qualquer outra possibilidade de gerenciamento de informações que não condizem com os requisitos levantados e descritos no tópico 3.1.3.

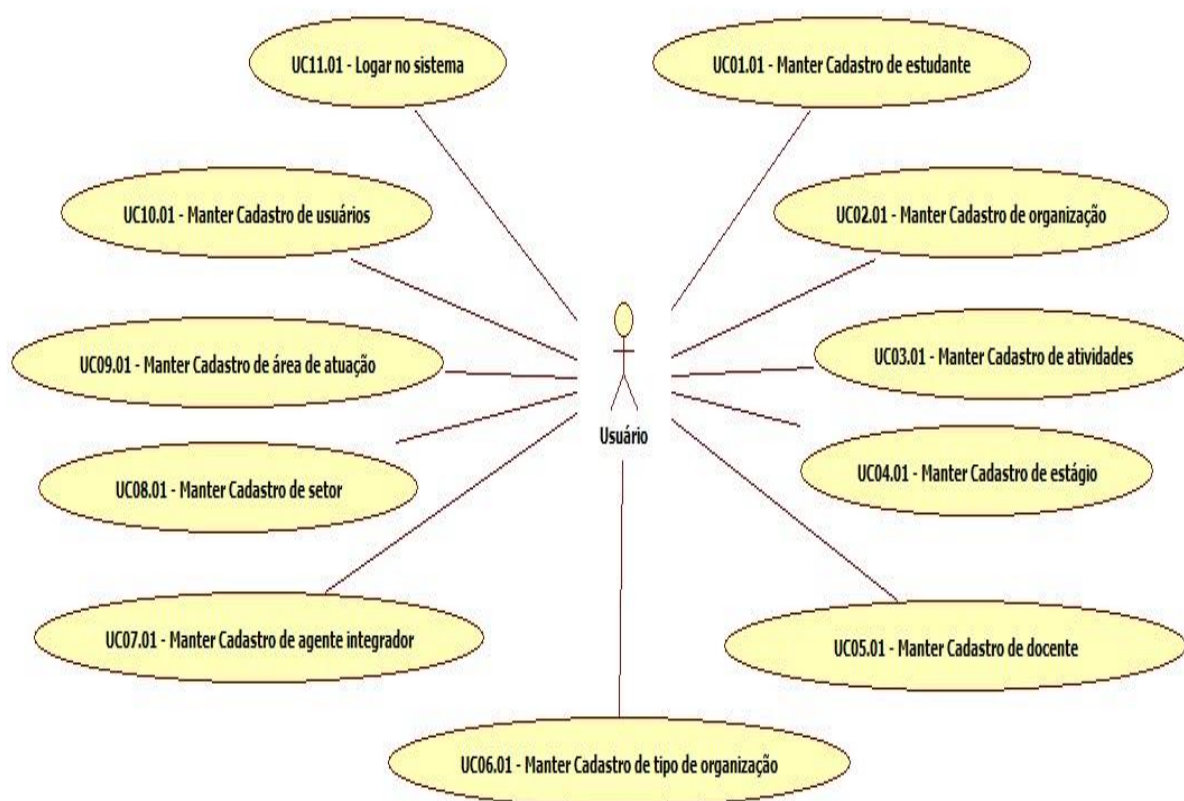


### 3.1.5. Especificação

A seguir serão mostrados os diagramas e modelos usados para o desenvolvimento e especificação do sistema.

#### 3.1.5.1. Diagramas de Casos de Uso

Com a finalidade de apresentar de forma simplificada o funcionamento do sistema e seus requisitos, os diagramas de caso de uso foram criados para facilitar o entendimento do cliente e a comunicação entre os envolvidos (BARRANCO DE AREBA, 2002). São apresentados nesta seção os casos de uso usados para desenvolvimento do sistema.



**Figura 2 - Diagrama de casos de uso.**

Fonte: Autor

### 3.1.5.2. Especificação de Casos de Uso

A especificação dos casos de uso é importante para facilitar a compreensão do sistema de forma detalhada, ele possibilita uma visão abrangente e completa da solução proposta.

<b>NOME:</b>	<b>MANTER CADASTRO DE ESTUDANTE</b>
<b>CASO DE USO</b>	<b>UC01.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de estudante no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados do estudante no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um estudante cadastrado e irá alterar o cadastro do estudante no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um estudante cadastrado e irá excluir o cadastro do estudante no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do estudante no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir o estudante selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de estudante com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um estudante.</b>
	C01.1 O usuário acessa o menu “Estudante” e clica no sub menu “Estudante cadastro”.
	C01.2 O sistema apresenta a página com o formulário de estudante e a tabela de consulta.
	C01.3 O usuário informa o GRR igual a “GRR20092222”.
	C01.4 O usuário informa o nome igual a “Jean”.
	C01.5 O usuário informa o e-mail igual a “jean@mail.com”.
	C01.6 O usuário informa o telefone residencial igual a “3333-3333”.
	C01.7 O usuário informa o telefone celular igual a “9999-9999”.
	C01.8 O usuário clica no botão "Cadastrar".
	C01.9 O sistema salva os dados do estudante no banco de dados.
	C01.10 O sistema atualiza a página com o formulário de

	estudante e a tabela de consulta.
	<b>C02.Usuário altera um estudante.</b>
	C02.1 O usuário acessa o menu “Estudante” e clica no sub menu “Estudante cadastro”.
	C02.2 O sistema apresenta a página com o formulário de estudante e a tabela de consulta.
	C02.3 O usuário escolhe um estudante na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados do estudante no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de estudante e a tabela de consulta.
	<b>C03.Usuário exclui um estudante.</b>
	C03.1 O usuário acessa o menu “Estudante” e clica no sub menu “Estudante cadastro”.
	C03.2 O sistema apresenta a página com o formulário de estudante e a tabela de consulta.
	C03.3 O usuário escolhe um estudante na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados do estudante do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de estudante e a tabela de consulta.

**Caso de Uso 1 - UC01.01 Manter cadastro de estudante**

<b>NOME:</b>	<b>MANTER CADASTRO DE ORGANIZAÇÃO</b>
<b>CASO DE USO</b>	<b>UC02.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de organização no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema, ter cadastrado Tipo de organização (UC06.01) e Área de atuação (UC09.01)
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados da organização no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma organização cadastrada e irá alterar o cadastro da organização no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma organização cadastrada e irá excluir o cadastro da organização no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	<b>A1.</b> O usuário não encontra o tipo de organização e precisa cadastrá-lo antes de cadastrar a organização.

<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados da organização no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir a organização selecionada.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de organização com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra uma organização.</b>
	C01.1 O usuário acessa o menu “Organização” e clica no sub menu “Organização cadastro”.
	C01.2 O sistema apresenta a página com o formulário de organização.
	C01.3 O usuário informa o CNPJ igual a “20.396.443/0001-50”.
	C01.4 O usuário informa o nome igual a “Empresa X”.
	C01.5 O usuário informa o e-mail igual a “empre@mail.com”.
	C01.6 O usuário informa o telefone igual a “3333-3333”.
	C01.7 O usuário informa o CEP igual a “81100-500”.
	C01.8 O usuário informa o logradouro igual a “Rua XYZ”.
	C01.9 O usuário informa o número igual a “450”.
	C01.10 O usuário informa o cidade igual a “Curitiba”.
	C01.11 O usuário informa o bairro igual a “Centro”.
	C01.12 O usuário informa o estado (UF) igual a “PR”.
	C01.13 O usuário informa o complemento igual a “2º andar”.
	C01.14 O usuário seleciona o tipo de organização igual a “Privada”.
	C01.15 O usuário seleciona a área de atuação igual a “Comércio”.
	C01.16 O usuário clica no botão "Cadastrar".
	C01.17 O sistema salva os dados da organização no banco de dados.
	C01.18 O sistema atualiza a página com o formulário de organização e a tabela de consulta.
	<b>C02.Usuário altera uma organização.</b>
	C02.1 O usuário acessa o menu “Organização” e clica no sub menu “Organização cadastro”.
	C02.2 O sistema apresenta a página com o formulário de organização e a tabela de consulta.
	C02.3 O usuário escolhe uma organização na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados da organização no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de

	organização e a tabela de consulta.
	<b>C03.Usuário exclui uma organização.</b>
	C03.1 O usuário acessa o menu “Organização” e clica no sub menu “Organização cadastro”.
	C03.2 O sistema apresenta a página com o formulário de organização e a tabela de consulta.
	C03.3 O usuário escolhe uma organização na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados do organização do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de organização e a tabela de consulta.

**Caso de Uso 2 - UC02.01 Manter cadastro de organização**

<b>NOME:</b>	<b>MANTER CADASTRO DE ATIVIDADES</b>
<b>CASO DE USO</b>	<b>UC03.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de atividade no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados da atividade no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma atividade cadastrada e irá alterar o cadastro da atividade no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma atividade cadastrada e irá excluir o cadastro da atividade no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados da atividade no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir a atividade selecionada.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de atividade com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra uma atividade.</b>
	C01.1 O usuário acessa o menu “Estágio” e clica no sub menu “Atividade”.
	C01.2 O sistema apresenta a página com o formulário de atividade e a tabela de consulta.
	C01.3 O usuário informa o nome igual a “Análise de Sistemas”.

	C01.4 O usuário informa a descrição igual a "Análise".
	C01.5 O usuário clica no botão "Cadastrar".
	C01.6 O sistema salva os dados da atividade no banco de dados.
	C01.7 O sistema atualiza a página com o formulário de atividade e a tabela de consulta.
	<b>C02.Usuário altera uma atividade.</b>
	C02.1 O usuário acessa o menu "Estágio" e clica no sub menu "Atividade".
	C02.2 O sistema apresenta a página com o formulário de atividade e a tabela de consulta.
	C02.3 O usuário escolhe uma atividade na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados da atividade no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de atividade e a tabela de consulta.
	<b>C03.Usuário exclui uma atividade.</b>
	C03.1 O usuário acessa o menu "Estágio" e clica no sub menu "Atividade".
	C03.2 O sistema apresenta a página com o formulário de atividade e a tabela de consulta.
	C03.3 O usuário escolhe uma atividade na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados da atividade do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de atividade e a tabela de consulta.

**Caso de Uso 3 - UC03.01 Manter cadastro de atividades**

<b>NOME:</b>	<b>MANTER CADASTRO DE ESTÁGIO</b>
<b>CASO DE USO</b>	<b>UC04.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de estágio no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema, ter cadastrado Estudante (UC01.01), Docente (UC05.01), Agente Integrador (UC07.01), Organização (UC02.01), Setor (UC08.01) e Atividades (UC03.01).
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	O usuário irá cadastrar os dados de estágio no sistema, caso não seja efetuado o cadastro será gerada a exceção E1. O usuário irá selecionar um estágio cadastrado e irá alterar o

	<p>cadastro do estágio no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um estágio cadastrado e irá excluir o cadastro do estágio no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	<b>A1.</b> O usuário não encontra o estudante, ou docente, ou agente integrador, ou organização, ou setor, ou atividades e precisa cadastrá-los antes de cadastrar o estágio.
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do estágio no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir o estágio selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de estágio com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um estágio.</b>
	C01.1 O usuário acessa o menu “Estágio” e clica no sub menu “Estágio cadastro”.
	C01.2 O sistema apresenta a página com o formulário de estágio e a tabela de consulta.
	C01.3 O usuário informa deixa o campo obrigatório sem marcar..
	C01.4 O usuário seleciona o estudante “Jean”.
	C01.5 O usuário informa a data de início igual a “01/01/2013”.
	C01.6 O usuário informa a data de fim prevista igual a “01/07/2013”.
	C01.7 O usuário informa a hora início igual a “13h00”.
	C01.8 O usuário informa a hora fim igual a “19h00”.
	C01.9 O usuário informa o valor da bolsa igual a “1000,00”.
	C01.10 O usuário seleciona o professor supervisor igual a “Simao”.
	C01.11 O usuário seleciona o professor da COE igual a “Egon”.
	C01.12 O usuário seleciona o agente integrador “CIEE”.
	C01.13 O usuário seleciona a organização “Empresa X”.
	C01.14 O usuário seleciona o setor “TI”.
	C01.15 O usuário informa o supervisor igual a “Joao”.
	C01.16 O usuário seleciona a atividade “Análise de Sistemas”.
	C01.17 O usuário clica no botão "Cadastrar".
	C01.18 O sistema salva os dados do estágio no banco de dados.
	C01.19 O sistema atualiza a página com o formulário de estágio e a tabela de consulta.
	<b>C02.Usuário altera um estágio.</b>
	C02.1 O usuário acessa o menu “Estágio” e clica no sub menu

	"Estágio cadastro".
	C02.2 O sistema apresenta a página com o formulário de estágio e a tabela de consulta.
	C02.3 O usuário escolhe um estágio na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados do estágio no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de estágio e a tabela de consulta.
	<b>C03.Usuário exclui um estágio.</b>
	C03.1 O usuário acessa o menu "Estágio" e clica no sub menu "Estágio cadastro".
	C03.2 O sistema apresenta a página com o formulário de estágio e a tabela de consulta.
	C03.3 O usuário escolhe um estágio na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados do estágio do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de estágio e a tabela de consulta.

**Caso de Uso 4 - UC04.01 Manter cadastro de estágio**

<b>NOME:</b>	<b>MANTER CADASTRO DE DOCENTE</b>
<b>CASO DE USO</b>	<b>UC05.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de docente no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados do docente no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um docente cadastrado e irá alterar o cadastro do docente no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um docente cadastrado e irá excluir o cadastro do docente no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do docente no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir o docente selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de docente com a tabela de consulta



	atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um docente.</b>
	C01.1 O usuário acessa o menu “Docente” e clica no sub menu “Docente cadastro”.
	C01.2 O sistema apresenta a página com o formulário de docente e a tabela de consulta.
	C01.3 O usuário informa a matrícula “25222”.
	C01.4 O usuário informa o nome igual a “Egon”.
	C01.5 O usuário informa o e-mail igual a “egon@mail.com”.
	C01.6 O usuário informa a sala igual a “22”.
	C01.7 O usuário informa o telefone da UFPR igual a “3333-2222”.
	C01.8 O usuário informa o telefone residencial igual a “5555-5555”.
	C01.9 O usuário informa o telefone celular igual a “9999-8888”.
	C01.10 O usuário informa que o docente é integrante da COE marcando o campo COE(S/N).
	C01.11 O usuário clica no botão "Cadastrar".
	C01.12 O sistema salva os dados do docente no banco de dados.
	C01.13 O sistema atualiza a página com o formulário de docente e a tabela de consulta.
	<b>C02.Usuário altera um docente.</b>
	C02.1 O usuário acessa o menu “Docente” e clica no sub menu “Docente cadastro”.
	C02.2 O sistema apresenta a página com o formulário de docente e a tabela de consulta.
	C02.3 O usuário escolhe um docente na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados do docente no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de docente e a tabela de consulta.
	<b>C03.Usuário exclui um docente.</b>
	C03.1 O usuário acessa o menu “Docente” e clica no sub menu “Docente cadastro”.
	C03.2 O sistema apresenta a página com o formulário de docente e a tabela de consulta.
	C03.3 O usuário escolhe um docente na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados do docente do banco de dados.

	C03.6 O sistema atualiza a página com o formulário de docente e a tabela de consulta.
--	---

**Caso de Uso 5 - UC05.01 Manter cadastro de docente**

<b>NOME:</b>	<b>MANTER CADASTRO DE TIPO DE ORGANIZAÇÃO</b>
<b>CASO DE USO</b>	<b>UC06.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de tipo de organização no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados de tipo de organização no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um tipo de organização cadastrado e irá alterar o cadastro do tipo de organização no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um tipo de organização cadastrado e irá excluir o cadastro do tipo de organização no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do tipo de organização no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir o tipo de organização selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de tipo de organização com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um tipo de organização.</b>
	C01.1 O usuário acessa o menu “Organização” e clica no sub menu “Tipo de Organização”.
	C01.2 O sistema apresenta a página com o formulário de tipo de organização e a tabela de consulta.
	C01.3 O usuário informa o tipo de organização igual a “Privada”.
	C01.4 O usuário clica no botão "Cadastrar".
	C01.5 O sistema salva os dados de tipo de organização no banco de dados.
	C01.6 O sistema atualiza a página com o formulário de tipo de organização e a tabela de consulta.
	<b>C02.Usuário altera um tipo de organização.</b>
	C02.1 O usuário acessa o menu “Organização” e clica no sub menu “Tipo de Organização”.
	C02.2 O sistema apresenta a página com o formulário de tipo

	de organização e a tabela de consulta.
	C02.3 O usuário escolhe um tipo de organização na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados de tipo de organização no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de tipo de organização e a tabela de consulta.
	<b>C03.Usuário exclui um tipo de organização.</b>
	C03.1 O usuário acessa o menu "Organização" e clica no sub menu "Tipo de Organização".
	C03.2 O sistema apresenta a página com o formulário de tipo de organização e a tabela de consulta.
	C03.3 O usuário escolhe um tipo de organização na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados de tipo de organização do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de tipo de organização e a tabela de consulta.

**Caso de Uso 6 - UC06.01 Manter cadastro de tipo de organização**

<b>NOME:</b>	<b>MANTER CADASTRO DE AGENTE INTEGRADOR</b>
<b>CASO DE USO</b>	<b>UC07.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de agente integrador no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados de agente integrador no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um agente integrador cadastrado e irá alterar o cadastro do agente integrador no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um agente integrador cadastrado e irá excluir o cadastro do agente integrador no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	<p>E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do agente integrador no sistema.</p> <p>E2. O sistema retorna a mensagem que não é possível excluir o</p>

	agente integrador selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de agente integrador com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um agente integrador.</b>
	C01.1 O usuário acessa o menu “Estágio” e clica no sub menu “Agente Integrador”.
	C01.2 O sistema apresenta a página com o formulário de agente integrador e a tabela de consulta.
	C01.3 O usuário informa o agente integrador igual a “CIEE”.
	C01.4 O usuário clica no botão "Cadastrar".
	C01.5 O sistema salva os dados de agente integrador no banco de dados.
	C01.6 O sistema atualiza a página com o formulário de agente integrador e a tabela de consulta.
	<b>C02.Usuário altera um agente integrador.</b>
	C02.1 O usuário acessa o menu “Estágio” e clica no sub menu “Agente Integrador”.
	C02.2 O sistema apresenta a página com o formulário de agente integrador e a tabela de consulta.
	C02.3 O usuário escolhe um agente integrador na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados de agente integrador no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de agente integrador e a tabela de consulta.
	<b>C03.Usuário exclui um agente integrador.</b>
	C03.1 O usuário acessa o menu “Estágio” e clica no sub menu “Agente Integrador”.
	C03.2 O sistema apresenta a página com o formulário de agente integrador e a tabela de consulta.
	C03.3 O usuário escolhe um agente integrador na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados de agente integrador do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de agente integrador e a tabela de consulta.

Caso de Uso 7 - UC07.01 Manter cadastro de agente integrador

<b>NOME:</b>	<b>MANTER CADASTRO DE SETOR</b>
<b>CASO DE USO</b>	<b>UC08.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de setor no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados de setor no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar um setor cadastrado e irá alterar o cadastro do setor no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar setor cadastrado e irá excluir o cadastro do setor no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados do setor no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir o setor selecionado.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de setor com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um setor.</b>
	C01.1 O usuário acessa o menu "Organização" e clica no sub menu "Setor".
	C01.2 O sistema apresenta a página com o formulário de setor e a tabela de consulta.
	C01.3 O usuário informa o setor igual a "TI".
	C01.4 O usuário clica no botão "Cadastrar".
	C01.5 O sistema salva os dados de setor no banco de dados.
	C01.6 O sistema atualiza a página com o formulário de setor e a tabela de consulta.
	<b>C02.Usuário altera um setor.</b>
	C02.1 O usuário acessa o menu "Organização" e clica no sub menu "Setor".
	C02.2 O sistema apresenta a página com o formulário de setor e a tabela de consulta.
	C02.3 O usuário escolhe um setor na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados de setor no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de setor e a tabela de consulta.
	<b>C03.Usuário exclui um setor.</b>

	C03.1 O usuário acessa o menu “Organização” e clica no sub menu “Setor”.
	C03.2 O sistema apresenta a página com o formulário de setor e a tabela de consulta.
	C03.3 O usuário escolhe um setor na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados de setor do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de setor e a tabela de consulta.

**Caso de Uso 8 - UC08.01 Manter cadastro de setor**

<b>NOME:</b>	<b>MANTER CADASTRO DE ÁREA DE ATUAÇÃO</b>
<b>CASO DE USO</b>	<b>UC09.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de área de atuação no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	<p>O usuário irá cadastrar os dados de área de atuação no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma área de atuação cadastrada e irá alterar o cadastro da área de atuação no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.</p> <p>O usuário irá selecionar uma área de atuação cadastrada e irá excluir o cadastro da área de atuação no sistema, caso não seja efetuada a exclusão será gerada a exceção E2.</p>
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra ou não altera os dados da área de atuação no sistema.
	E2. O sistema retorna a mensagem que não é possível excluir a área de atuação selecionada.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de área de atuação com a tabela de consulta atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra uma área de atuação.</b>
	C01.1 O usuário acessa o menu “Organização” e clica no sub menu “Área de Atuação”.
	C01.2 O sistema apresenta a página com o formulário de área de atuação e a tabela de consulta.
	C01.3 O usuário informa a área de atuação igual a “Comércio”.
	C01.4 O usuário clica no botão "Cadastrar".
	C01.5 O sistema salva os dados de área de atuação no banco de dados.

	C01.6 O sistema atualiza a página com o formulário de área de atuação e a tabela de consulta.
	<b>C02.Usuário altera uma área de atuação.</b>
	C02.1 O usuário acessa o menu “Organização” e clica no sub menu “Área de Atuação”.
	C02.2 O sistema apresenta a página com o formulário de área de atuação e a tabela de consulta.
	C02.3 O usuário escolhe uma área de atuação na tabela de consulta.
	C02.4 O usuário altera as informações necessárias.
	C02.5 O usuário clica no botão "Alterar".
	C02.6 O sistema salva os dados de área de atuação no banco de dados.
	C02.7 O sistema atualiza a página com o formulário de área de atuação e a tabela de consulta.
	<b>C03.Usuário exclui uma área de atuação.</b>
	C03.1 O usuário acessa o menu “Organização” e clica no sub menu “Área de Atuação”.
	C03.2 O sistema apresenta a página com o formulário de área de atuação e a tabela de consulta.
	C03.3 O usuário escolhe uma área de atuação na tabela de consulta.
	C03.4 O usuário clica no botão "Deletar".
	C03.5 O sistema exclui os dados de área de atuação do banco de dados.
	C03.6 O sistema atualiza a página com o formulário de área de atuação e a tabela de consulta.

Caso de Uso 9 - UC09.01 Manter cadastro de área de atuação

<b>NOME:</b>	<b>MANTER CADASTRO DE USUÁRIOS</b>
<b>CASO DE USO</b>	<b>UC10.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por manter o cadastro de usuário no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário deve estar logado no sistema.
<b>ATOR:</b>	Todos os usuários com acesso ao sistema.
<b>FLUXO PRINCIPAL:</b>	O usuário irá cadastrar os dados do usuário no sistema, caso não seja efetuado o cadastro será gerada a exceção E1.
<b>FLUXO ALTERNATIVO:</b>	
<b>FLUXO DE EXCEÇÃO:</b>	E1. Usuário não preenche os campos obrigatórios e o sistema não cadastra o usuário no sistema.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela de usuário atualizada.
<b>CENÁRIO</b>	<b>C01.Usuário cadastra um usuário.</b>

	C01.1 O usuário acessa o menu “Configurações” e clica no sub menu “Usuário cadastro”.
	C01.2 O sistema apresenta a página com o formulário de usuário.
	C01.3 O usuário informa o nome do usuário igual a “Jean”.
	C01.4 O usuário informa uma senha para o usuário igual a “Jean”.
	C01.5 O usuário clica no botão "Cadastrar".
	C01.6 O sistema salva os dados do usuário no banco de dados.
	C01.7 O sistema atualiza a página com o formulário de usuário.

Caso de Uso 10 - UC10.01 Manter cadastro de usuários

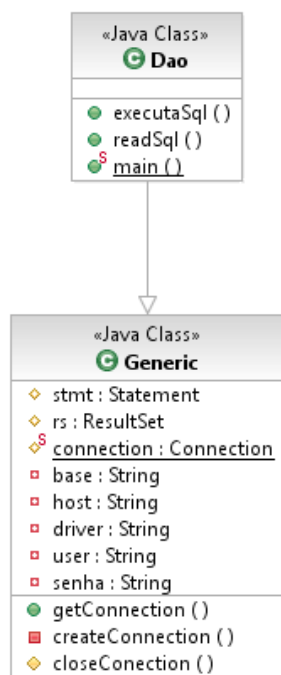
<b>NOME:</b>	<b>LOGAR NO SISTEMA</b>
<b>CASO DE USO</b>	<b>UC11.01</b>
<b>DESCRIÇÃO:</b>	Esse caso de uso descreve o processo responsável por efetuar o <i>login</i> do usuário no sistema.
<b>PRÉ-CONDIÇÃO:</b>	O usuário de estar cadastrado no sistema.
<b>ATOR:</b>	Todos os usuários com cadastro no sistema.
<b>FLUXO PRINCIPAL:</b>	O usuário entrará no sistema, caso não seja efetuado o <i>login</i> será gerada as exceções E1 e E2.
<b>FLUXO ALTERNATIVO:</b>	<b>A1.</b> Caso o usuário não tenha acesso, ele deve fazer uma requisição a outro usuário com acesso ao sistema.
<b>FLUXO DE EXCEÇÃO:</b>	E1. O <i>login</i> ou a senha estão incorretos.
<b>PÓS-CONDIÇÃO</b>	É apresentada a tela inicial.
<b>CENÁRIO</b>	<b>C01.Usuário faz <i>login</i> no sistema.</b>
	C01.1 O usuário acessa o sistema através do endereço <u>localhost:8080/decigi/Coe.html</u> .
	C01.2 O sistema apresenta a página com os campos para o <i>login</i> .
	C01.3 O usuário informa o <i>login</i> igual a “Jean”.
	C01.4 O usuário informa a senha igual a “Jean”.
	C01.5 O usuário clica no botão " <i>Login</i> ".
	C01.6 O sistema redireciona o sistema para a tela principal.

Caso de Uso 11 - UC11.01 Logar no sistema



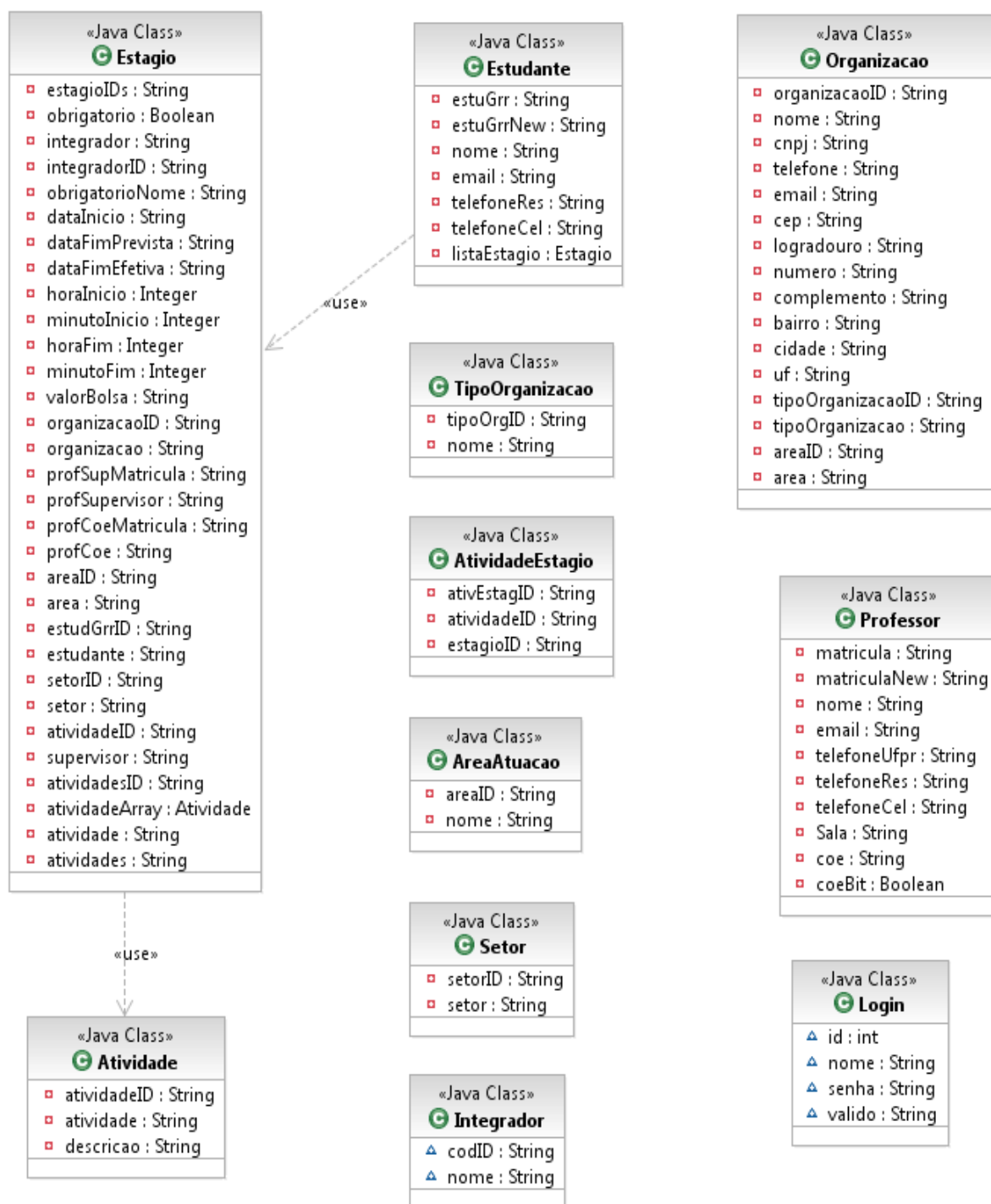
### 3.1.5.3. Diagrama de Classe

Abaixo serão apresentados os diagramas de classe usados para desenvolvimento do sistema.



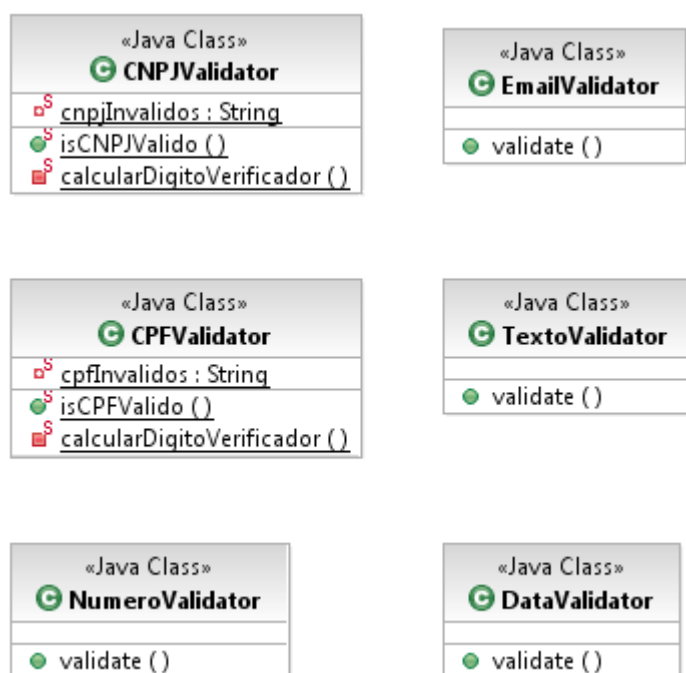
**Figura 3 – Pacote “Dao”**

Fonte: Autor

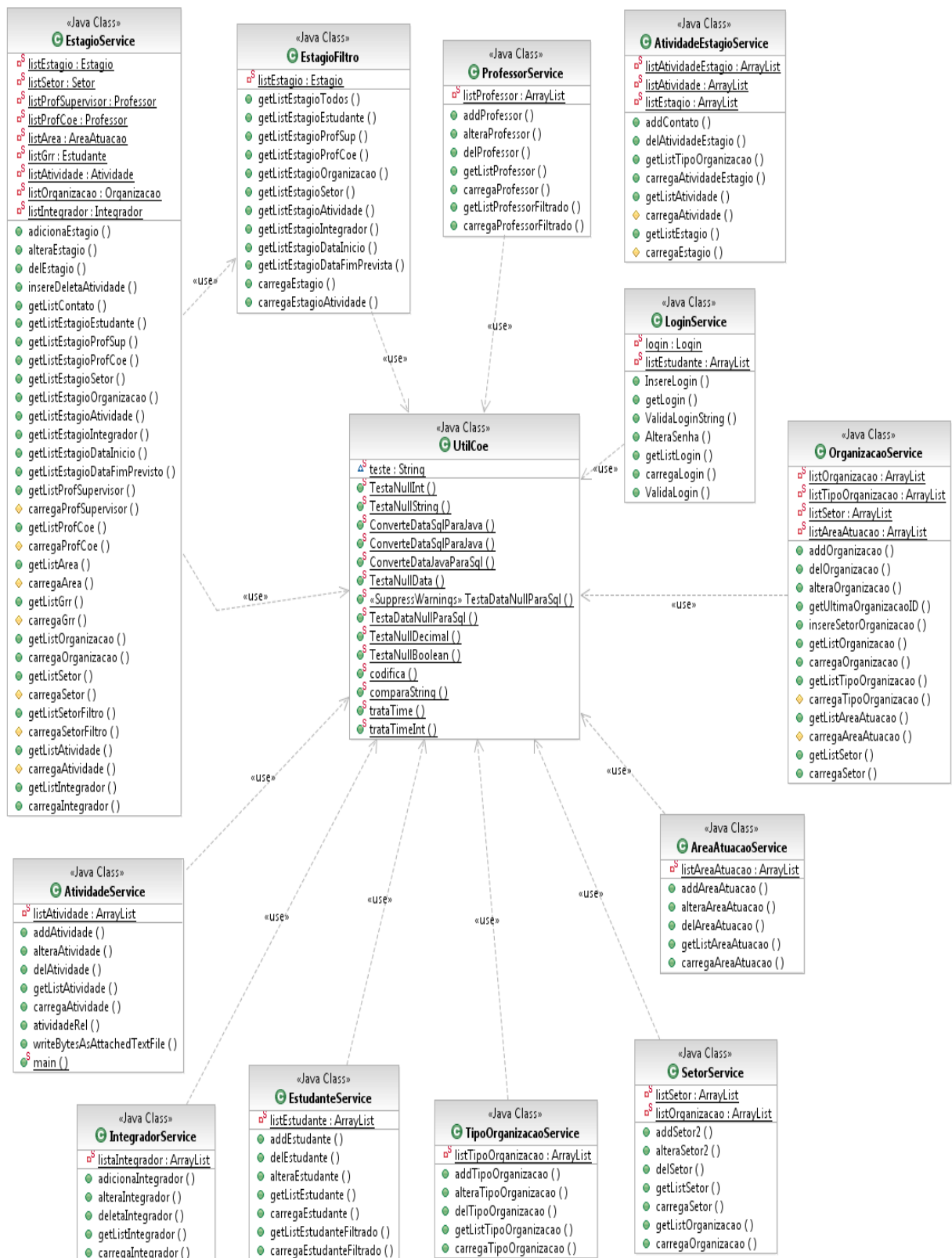


**Figura 4 – Pacote “Model”**

Fonte: Autor



**Figura 5 - Pacote "Validator"**  
Fonte: Autor

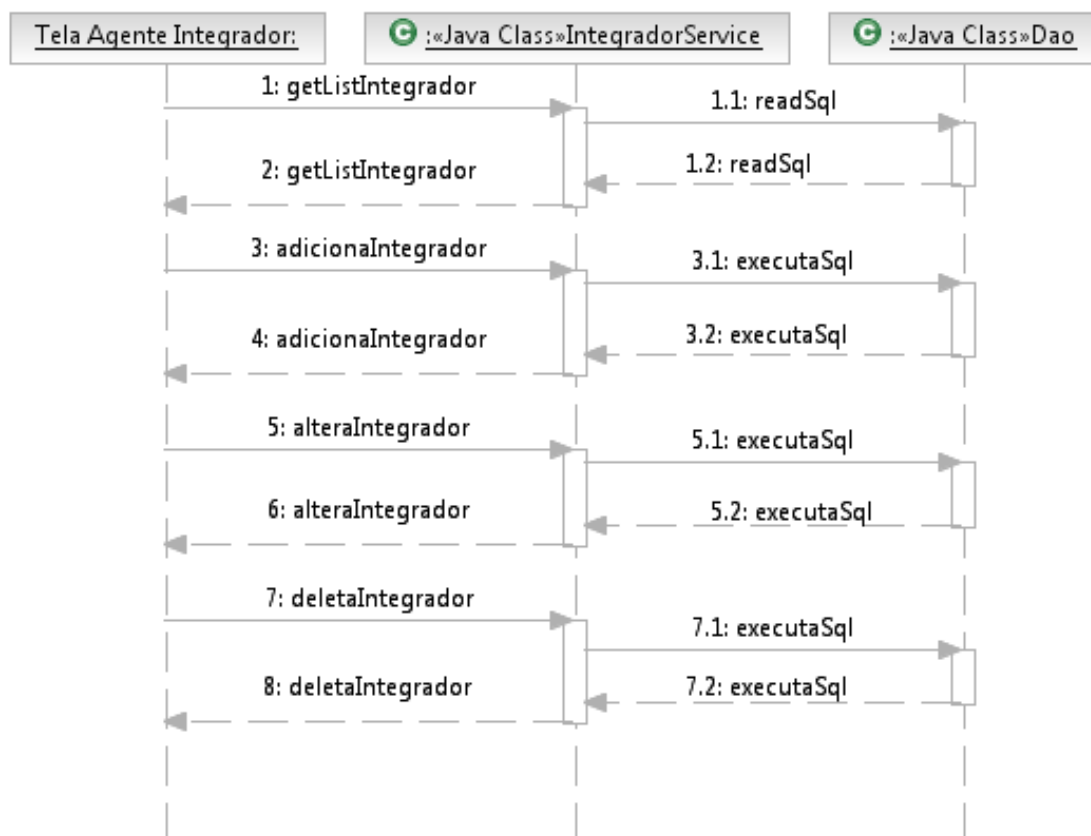


**Figura 6 - Pacote "Service"**

Fonte: Autor

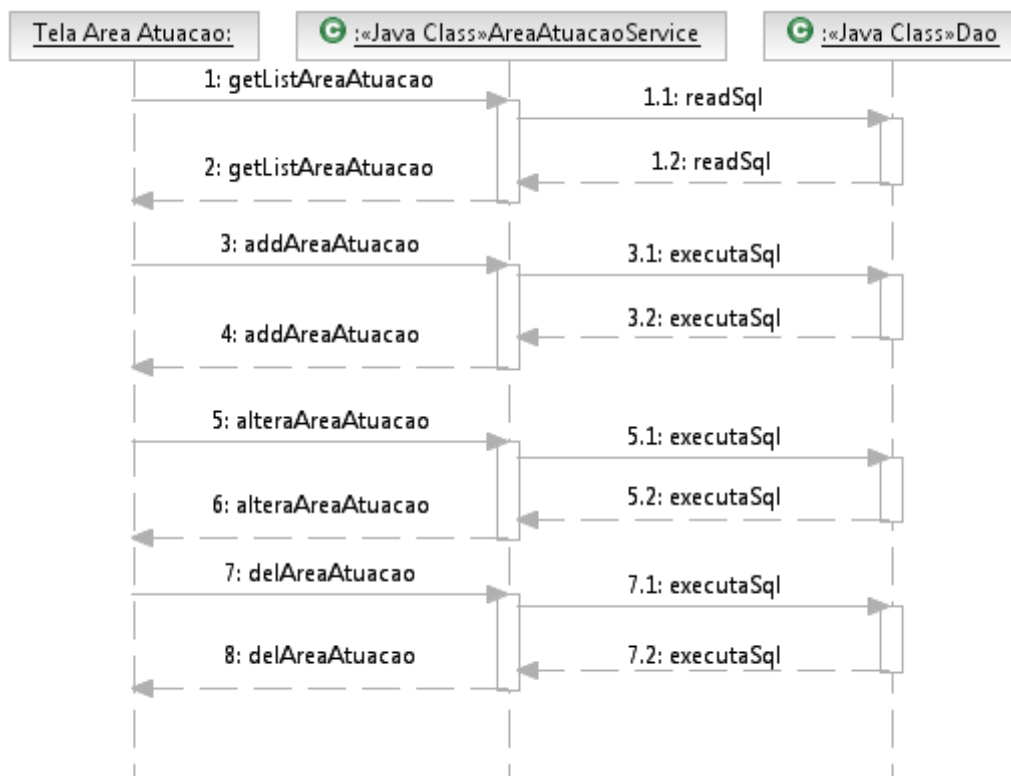
#### 3.1.5.4. Diagrama de Sequência

A seguir são mostrados os diagramas de sequência que descrevem a sequência lógica e temporal do negócio.



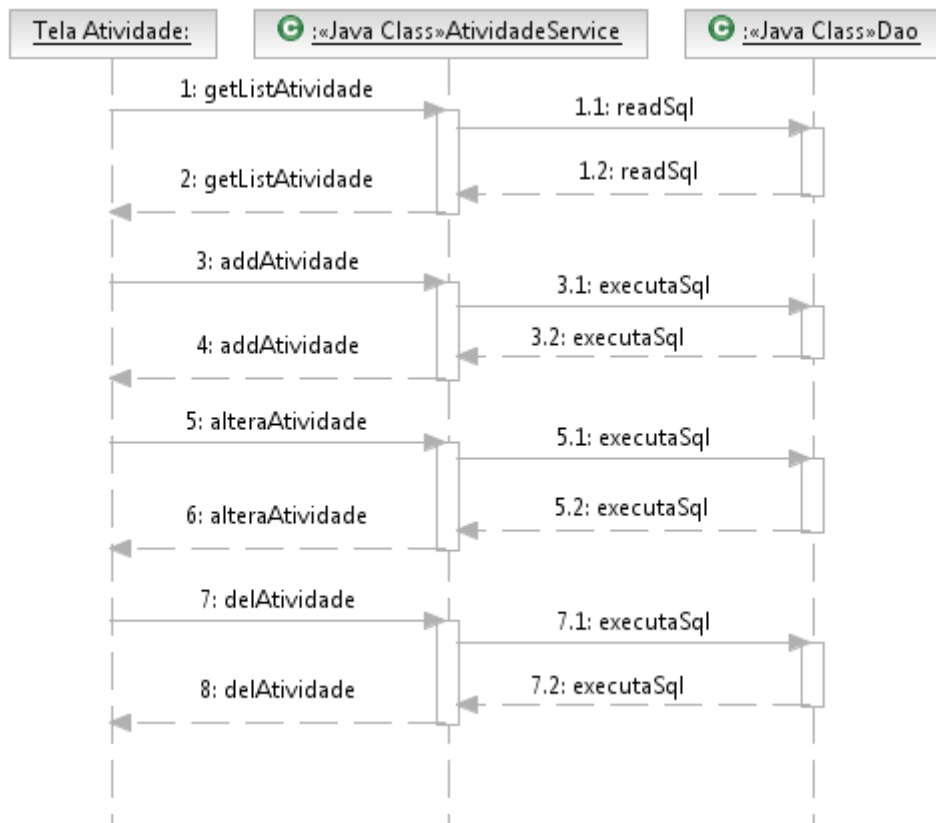
**Figura 7 - Diagrama de Sequência "AgenteIntegrador"**

Fonte: Autor



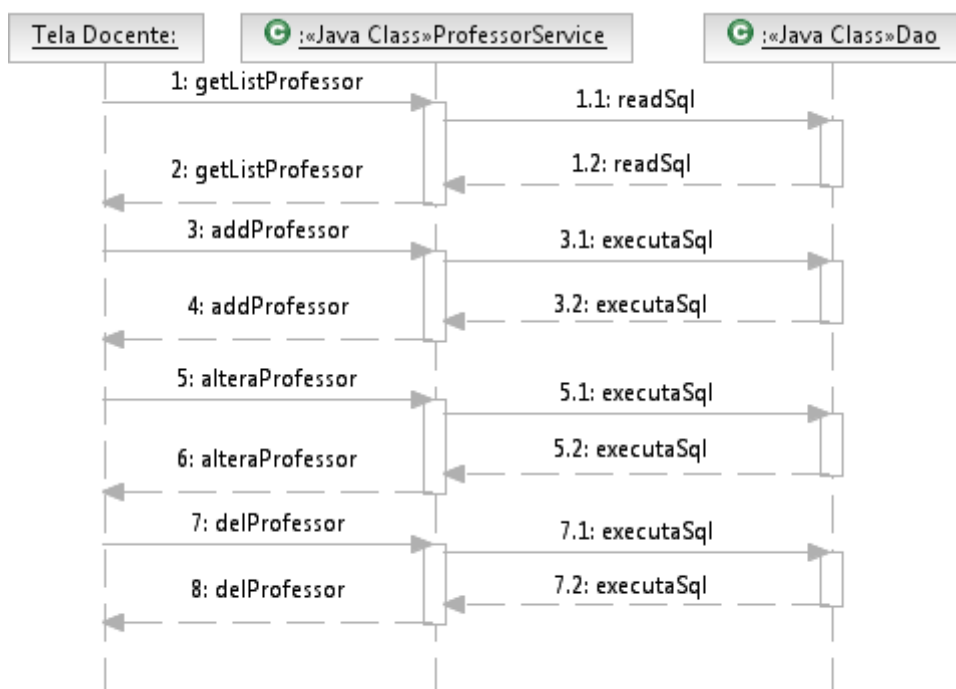
**Figura 8 - Diagrama de Sequência "AreaAtuacao"**

Fonte: Autor



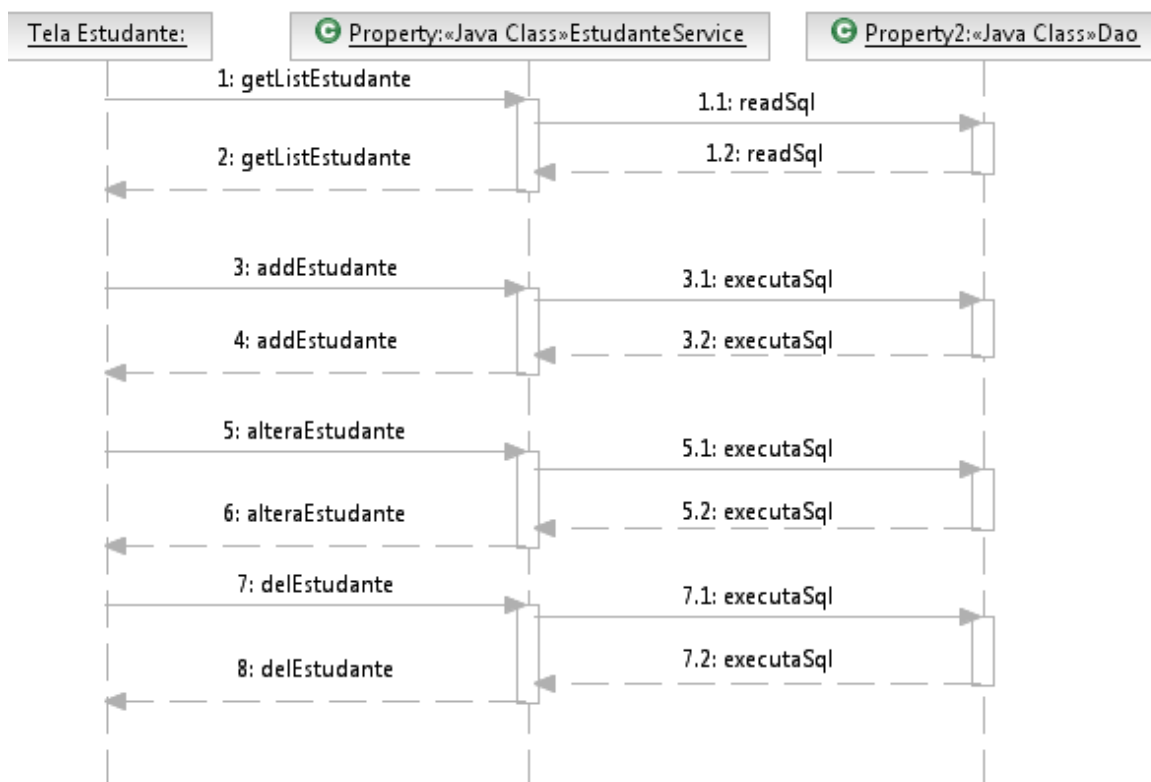
**Figura 9 - Diagrama de Sequência "Atividade"**

Fonte: Autor



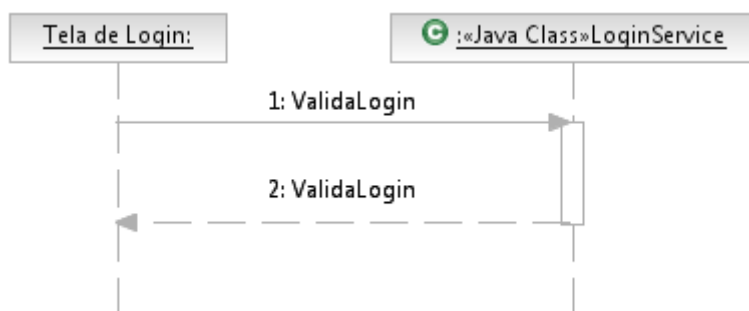
**Figura 10 - Diagrama de Sequência "Docente"**

Fonte: Autor



**Figura 11 - Diagrama de Sequência "Estudante"**

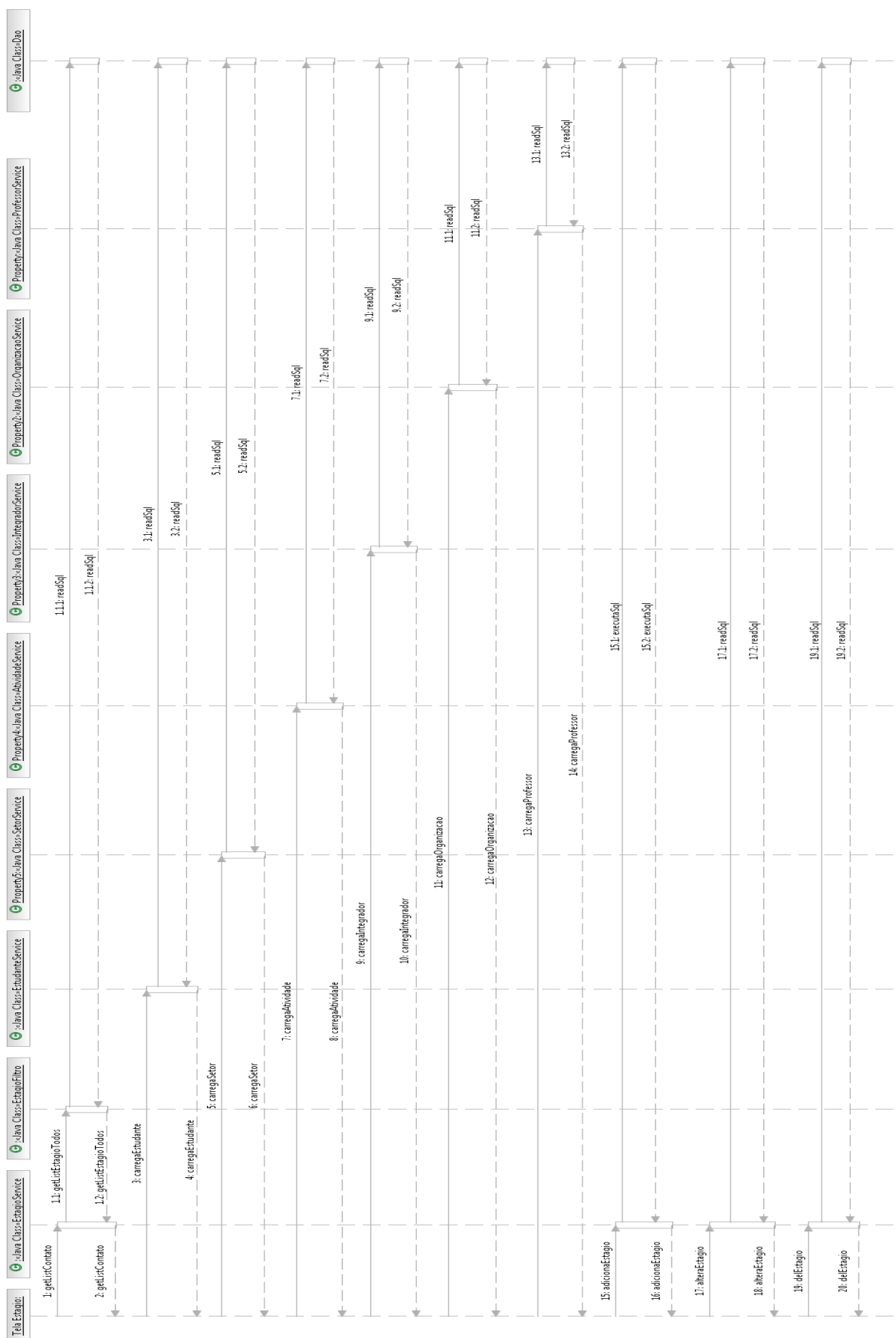
Fonte: Autor



**Figura 12 - Diagrama de Sequência "Login"**

Fonte: Autor





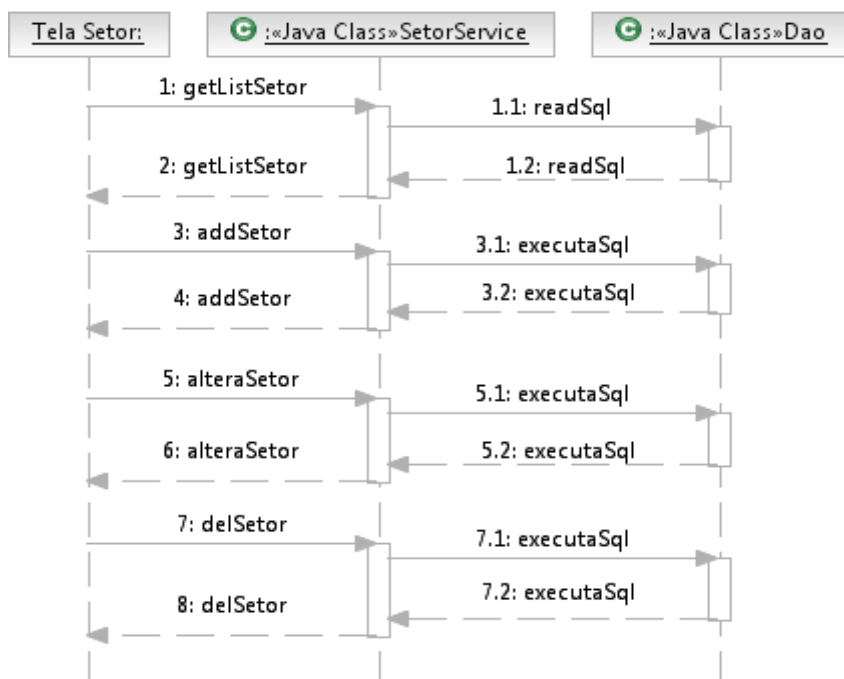
**Figura 13 - Diagrama de Sequência "Estagio"**

Fonte: Autor



**Figura 14 - Diagrama de Sequência "Organizacao"**

Fonte: Autor



**Figura 15 - Diagrama de Sequência "Setor"**

Fonte: Autor

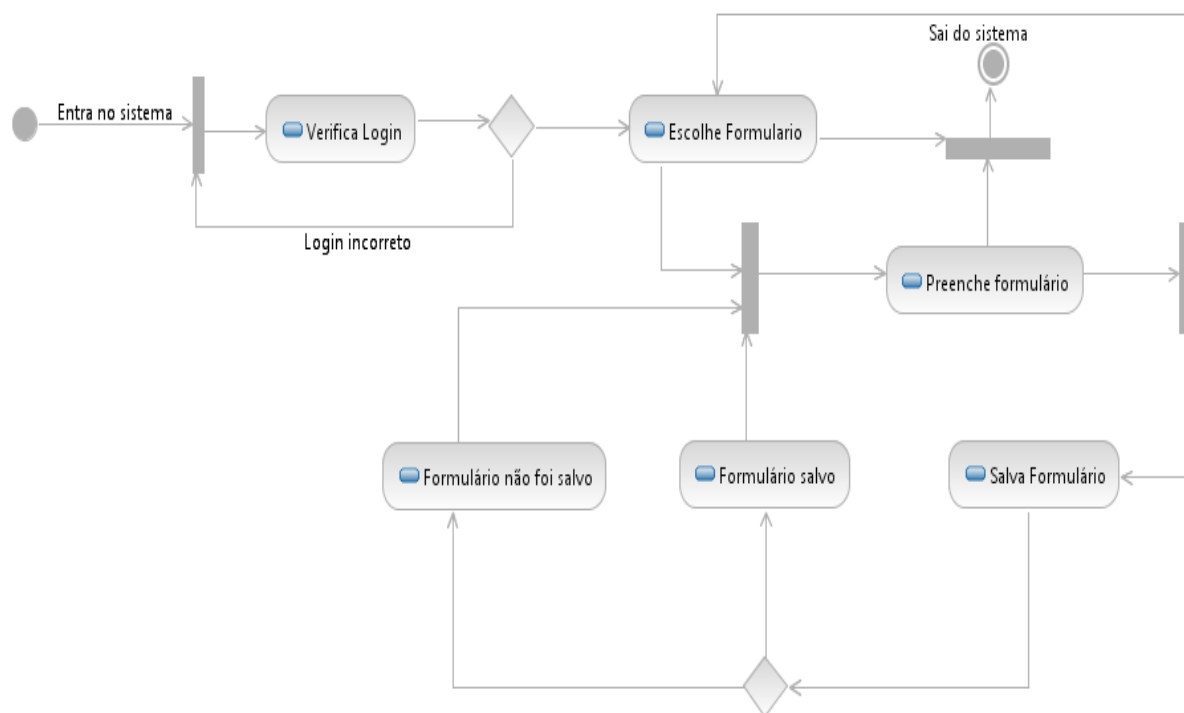


**Figura 16 - Diagrama de Sequência "TipoOrganizacao"**

Fonte: Autor

### 3.1.5.5. Diagrama de Atividade

Abaixo é apresentado o diagrama de atividade que visa representar o fluxo dos processos do sistema.



**Figura 17 - Diagrama de Atividade do sistema**

Fonte: Autor

### 3.1.5.6. Diagrama de Estado

Abaixo são mostrados os diagramas que tem por finalidade mostrar quais os estados que o objeto pode ter no sistema.



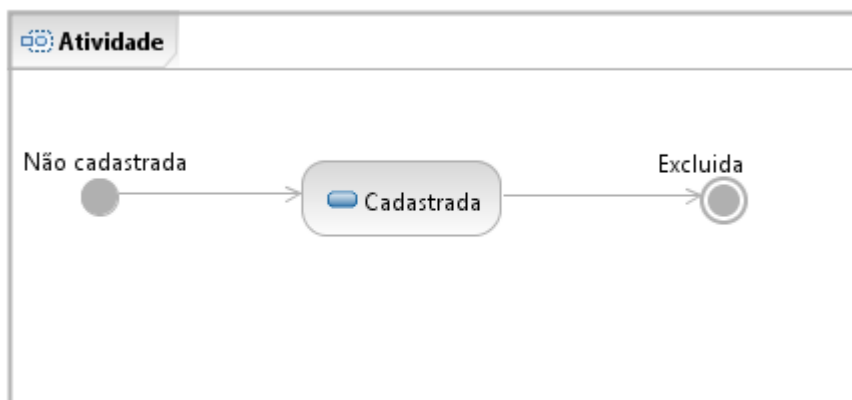
**Figura 18 - Diagrama de estado "AgenteIntegrador"**

Fonte: Autor



**Figura 19 - Diagrama de estado "AreaAtuacao"**

Fonte: Autor



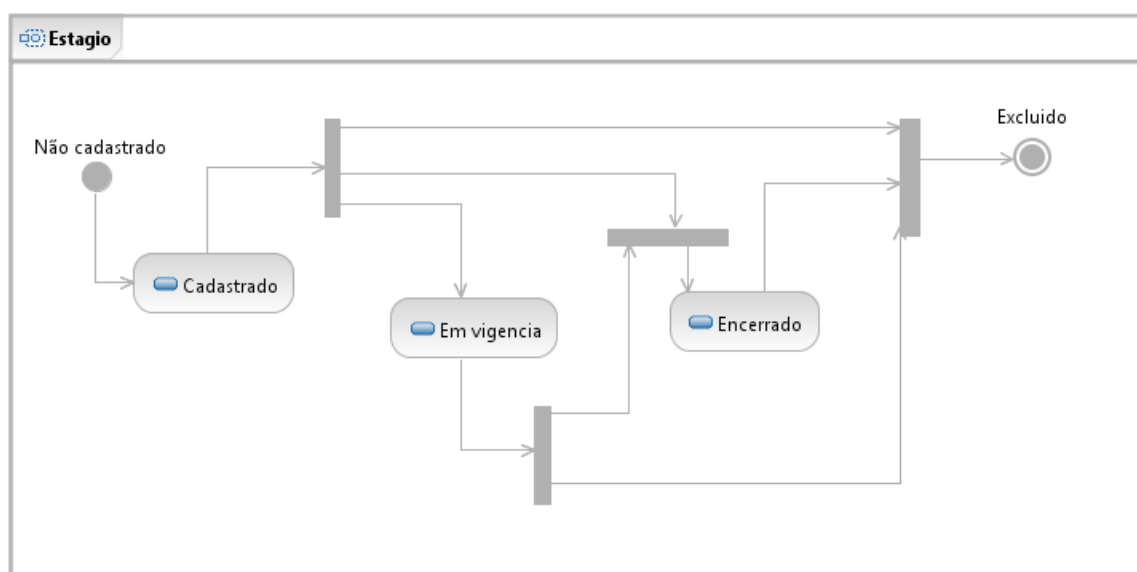
**Figura 20 - Diagrama de estado "Atividade"**

Fonte: Autor



**Figura 21 - Diagrama de estado "Docente"**

Fonte: Autor



**Figura 22 - Diagrama de estado "Estagio"**

Fonte: Autor



**Figura 23 - Diagrama de estado "Estudante"**

Fonte: Autor



**Figura 24 - Diagrama de estado "Organizacao"**

Fonte: Autor



**Figura 25 - Diagrama de estado "Setor"**

Fonte: Autor

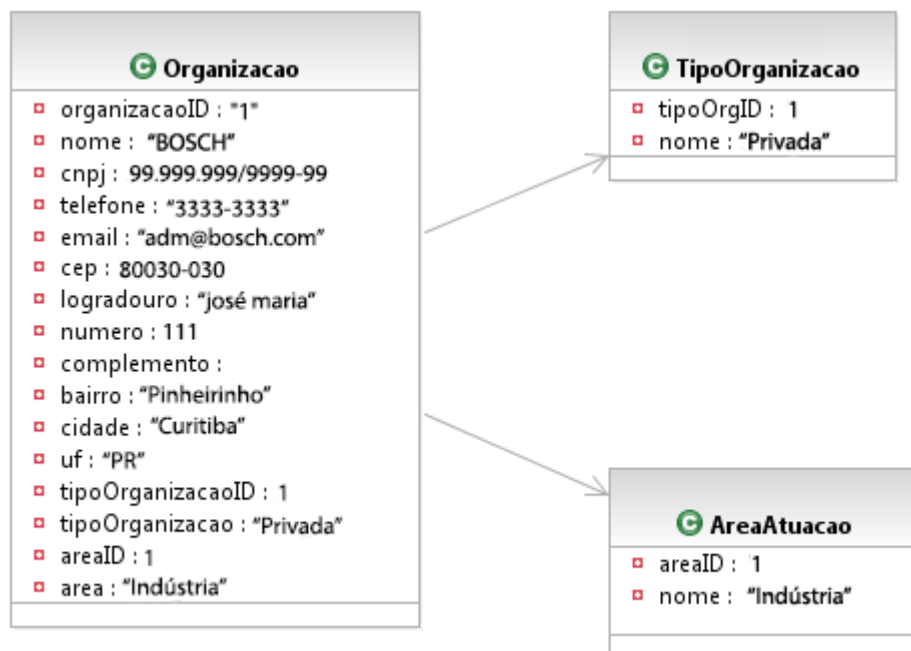


**Figura 26 - Diagrama de estado "TipoOrganizacao"**

Fonte: Autor

### 3.1.5.7. Diagrama de Objeto

O diagrama apresentado a seguir demonstra uma instância de um objeto no sistema.



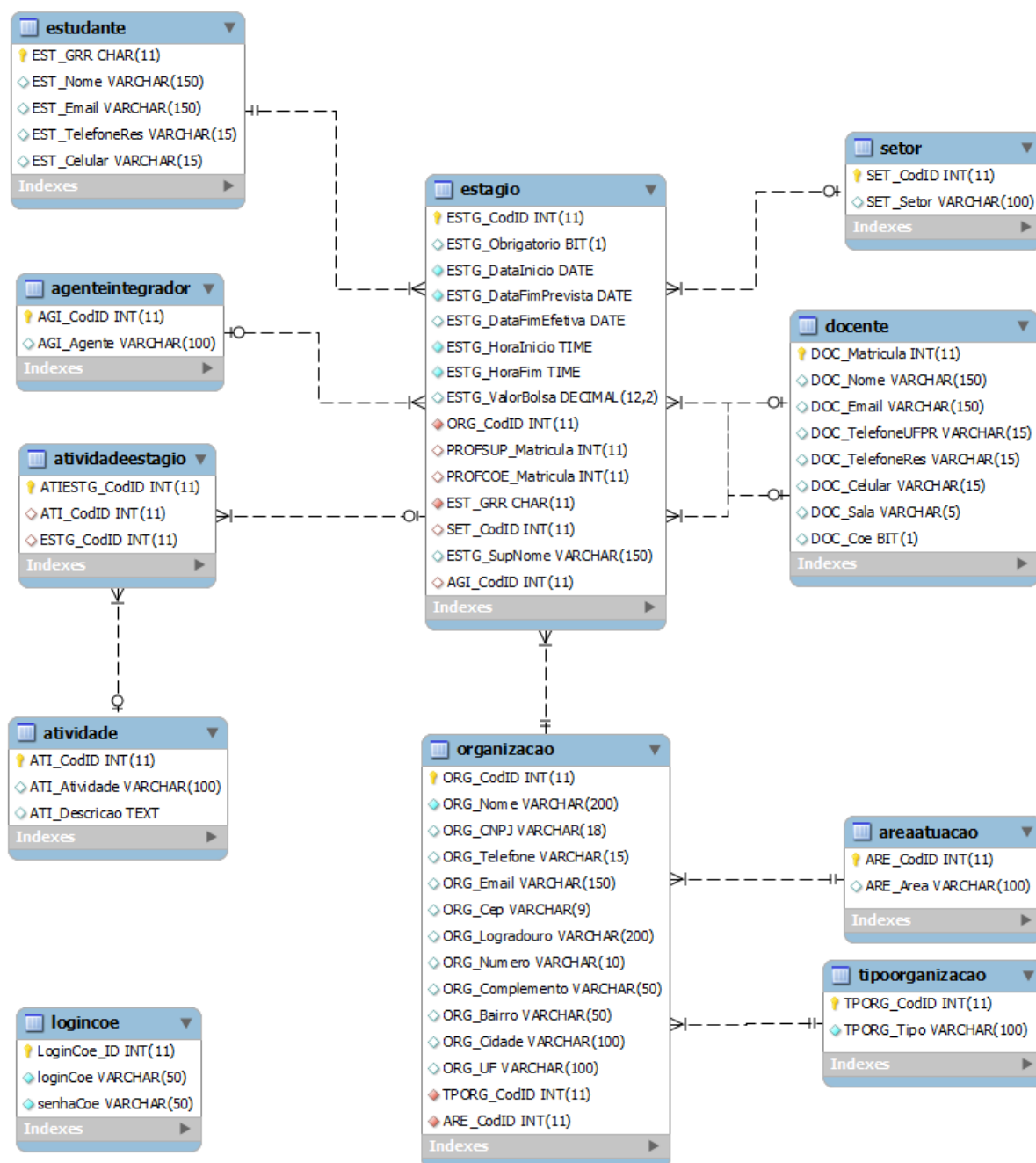
**Figura 27 - Diagrama de objeto "Organizacao"**

Fonte: Autor



### 3.1.5.8. Diagrama Entidade-Relacionamento

A figura 4 apresenta o diagrama entidade-relacionamento (DER) que representa a estrutura do banco de dados criado para atender os requisitos abordados no capítulo 3.1.3.



**Figura 28 - Diagrama Entidade – Relacionamento**

Fonte: Autor

### 3.1.5.9. Dicionário de Dado e Modelo de Tabelas Normalizado

O dicionário de dados tem a finalidade de apresentar informações sobre as entidades e seus atributos, pois apenas o DER não mostra todas as informações necessárias para a criação do banco de dados. Abaixo é apresentado o dicionário de dados do sistema.

Estudante								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
EST_GRR	Matrícula do estudante na UFPR.	CHAR	11	Sim	Não	Sim	Não	
EST_Nome	Nome do estudante.	VARCHAR	150	Não	Não	Não	Não	
EST_Email	E-mail do estudante.	VARCHAR	150	Não	Não	Não	Sim	
EST_TelefoneRes	Telefone residencial do estudante.	VARCHAR	15	Não	Não	Não	Sim	
EST_Celular	Telefone celular do estudante.	VARCHAR	15	Não	Não	Não	Sim	

**Tabela 3 - Especificação da tabela "Estudante"**

Docente								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
DOC_Matricula	Matriculo do docente na UFPR.	INT	11	Sim	Não	Sim	Não	
DOC_Nome	Nome do docente	VARCHAR	150	Não	Não	Não	Não	
DOC_Email	E-mail do docente.	VARCHAR	150	Não	Não	Não	Sim	
DOC_Telefone UFPR	Telefone da sala do docente.	VARCHAR	15	Não	Não	Não	Sim	
DOC_Telefone Res	Telefone residencial do docente.	VARCHAR	15	Não	Não	Não	Sim	
DOC_Celular	Telefone celular do docente	VARCHAR	15	Não	Não	Não	Sim	
DOC_Sala	Número da sala do docente.	VARCHAR	5	Não	Não	Não	Sim	
DOC_Coe	Indica se o docente faz parte da COE.	BIT	1	Não	Não	Não	Não	Se o valor do campo for zero indica que o docente não faz parte da COE, se o campo for um indica que o docente é da COE.

Tabela 4 - Especificação da tabela "Docente"

Setor								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
SET_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
SET_Setor	Nome do setor.	VARCHAR	100	Não	Não	Sim	Não	Setores que fazem parte de uma organização.

Tabela 5 - Especificação da tabela "Setor"

Areaatuacao								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
ARE_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
ARE_Area	Área de atuação.	VARCHAR	100	Não	Não	Sim	Não	Indica a área de atuação das empresas. Ex. Comércio, Indústria.

Tabela 6 - Especificação da tabela "Areaatuacao"

Atividade								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
ATI_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
ATI_Atividade	Nome da atividade.	VARCHAR	100	Não	Não	Sim	Não	Atividades desenvolvidas pelos estudantes nos estágios.
ATI_Descricao	Descrição da atividade.	TEXT		Não	Não	Não	Sim	

Tabela 7 - Especificação da tabela "Atividade"

Atividadeestagio								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
ATIESTG_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
ATI_CodID	Código da tabela de atividade.	INT	11	Não	Sim	Não	Não	
ESTG_CodID	Código da tabela de estagio.	INT	11	Não	Sim	Não	Não	

Tabela 8 - Especificação da tabela "Atividadeestagio"

Agenteintegrador								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
AGI_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
AGI_Agente	Nome do agente integrador.	VARCHAR	100	Não	Não	Sim	Não	Ex. CIEE, IEL.

Tabela 9 - Especificação da tabela "Agenteintegrador"

Organizacao								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
ORG_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
ORG_Nome	Nome da organização.	VARCHAR	200	Não	Não	Não	Não	
ORG_CNPJ	CNPJ da organização.	CHAR	18	Não	Não	Sim	Sim	
ORG_Telefone	Telefone da organização.	VARCHAR	15	Não	Não	Não	Sim	
ORG_Email	E-mail da organização.	VARCHAR	150	Não	Não	Não	Sim	
ORG_Cep	CEP onde a organização está localizada.	VARCHAR	9	Não	Não	Não	Sim	
ORG_Logradouro	Logradouro onde a organização está localizada..	VARCHAR	200	Não	Não	Não	Sim	
ORG_Numero	Número onde a organização está	VARCHAR	10	Não	Não	Não	Sim	

	localizada.							
ORG_Complemento	Complemento de onde a organização esta localizada.	VARCHAR	50	Não	Não	Não	Sim	Ex. 4º andar, sala 202.
ORG_Bairro	Bairro onde a organização esta localizada.	VARCHAR	50	Não	Não	Não	Sim	
ORG_Cidade	Cidade onde a organização esta localizada.	VARCHAR	100	Não	Não	Não	Sim	
ORG_UF	Estado onde a organização esta localizada.	VARCHAR	100	Não	Não	Não	Sim	
TPORG_CodID	Código do tipo de organização.	INT	11	Não	Sim	Não	Não	
ARE_CodID	Código da área de atuação da organização.	INT	11	Não	Sim	Não	Não	

Tabela 10 - Especificação da tabela "Organizacao"

Tipoorganizacao								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
TPORG_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
TPORG_Tipo	Tipo de organização.	VARCHAR	100	Não	Não	Sim	Não	

Tabela 11 - Especificação da tabela "Tipoorganizacao"

Estagio								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
ESTG_CodID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
ESTG_Obrigatorio	Indica se o estagio é obrigatório ou não.	BIT	1	Não	Não	Não	Não	Se o valor do campo for zero o estágio não é obrigatório, se o valor for um o estágio é obrigatório.
ESTG_DataInicio	Data de início do estágio.	DATE		Não	Não	Não	Não	
ESTG_DataFimPrevista	Data de fim prevista para o estágio no contrato.	DATE		Não	Não	Não	Não	
ESTG_DataFimEfetiva	Data de fim efetiva do estágio.	DATE		Não	Não	Não	Sim	
ESTG_HoraInicio	Horário de início do estágio.	TIME		Não	Não	Não	Não	
ESTG_HoraFim	Horário de fim do estágio.	TIME		Não	Não	Não	Não	
ESTG_ValorBolsa	Valor da bolsa que o estudante receberá.	DECIMAL	12,2	Não	Não	Não	Sim	
ESTG_SupNome	Nome do supervisor.	VARCHAR	150	Não	Não	Não	Sim	Nome do supervisor do estudante na organização.
ORG_CodID	Código da organização.	INT	11	Não	Sim	Não	Não	
PROFSUP_Matricula	Matrícula do docente supervisor.	INT	11	Não	Sim	Não	Sim	
PROFCOE_Matricula	Matrícula do docente da COE.	INT	11	Não	Sim	Não	Sim	Matricula do docente responsável pelo estágio.
EST_GRR	Matrícula do estudante.	CHAR	11	Não	Sim	Não	Não	
SET_CodID	Código do setor.	INT	11	Não	Sim	Não	Sim	
AGI_CodID	Código do agente integrador.	INT	11	Não	Sim	Não	Sim	

Tabela 12- Especificação da tabela "Estagio"

LoginCoe								
Atributo	Descrição	Tipo	Tamanho	PK	FK	ÚNICO	NULO	OBS.
loginCoe_ID	Identificador da tabela.	INT	11	Sim	Não	Sim	Não	
loginCoe	Login .	VARCHAR	50	Não	Não	Sim	Não	Login usado para acessar o sistema.
senhaCoe	Senha.	VARCHAR	50	Não	Não	Não	Não	Senha criptografada usada para acessar o sistema.

Tabela 13 - Especificação da tabela "LoginCoe"



### 3.1.6. Implementação

Nos tópicos a seguir serão apresentadas as técnicas e ferramentas utilizadas para o desenvolvimento do sistema e as operacionalidades do sistema.

#### 3.1.6.1. Técnicas e Ferramentas Utilizadas

O sistema foi desenvolvido em *JAVA* e *FLEX*, as ferramentas usadas foram *Eclipse* e *Adobe Flash Builder* respectivamente que permitem a manipulação de códigos em *JAVA* e *FLEX*, também foi adotado o servidor web *Apache Tomcat 6.0* que permite a execução aplicações Java para web e usado o gerador de relatórios *BIRT (Business Intelligence and Reporting Tools)*.

O banco de dados escolhido foi o *MYSQL* na versão 5.5, pois é um sistema gratuito, o desenvolvedor já tem experiência com o BD e segundo Niederauer e Prates (2006) permite um número de acesso simultâneos ilimitado, tem uma alta velocidade de execução dos comandos, e uma grande quantidade de registros, mais de 50.000.000. O programa para manipular o banco de dados foi o *MySQL Workbench CE 5.2.34.2*.

A figura 4 apresenta um trecho do código *JAVA* usado para manipular os dados do docente e a figura 5 mostra um trecho do código *FLEX* para manipular os dados do estudante.

```
/**
 * Adiciona um docente
 * @param professor
 */
public void addProfessor(Professor professor) {

    String matricula = UtilCoe.TestaNullInt(professor.getMatricula());
    String nome = UtilCoe.TestaNullString(professor.getNome());
    String email = UtilCoe.TestaNullString(professor.getEmail());
    String telefoneUfpr = UtilCoe.TestaNullString(professor.getTelefoneUfpr());
    String telefoneRes = UtilCoe.TestaNullString(professor.getTelefoneRes());
    String telefoneCel = UtilCoe.TestaNullString(professor.getTelefoneCel());
    String sala = UtilCoe.TestaNullString(professor.getSala());
    String coe = UtilCoe.TestaNullInt(professor.getCoe());

    Dao dao = new Dao();
    String query = "INSERT INTO `coe`.`docente` (`DOC_Matricula`, `DOC_Nome`, `DOC_Email`, `DOC_TelefoneUfpr`, `DOC_TelefoneRes`, `DOC_TelefoneCel`, `DOC_Sala`, `DOC_Coe`) VALUES (" + matricula + ", " + nome + ", " + email + ", " + telefoneUfpr + ", " + telefoneRes + ", " + telefoneCel + ", " + sala + ", " + coe + ")";

    try{
        dao.executaSql(query);
    }
```

**Figura 29 - Código JAVA**

Fonte: Autor

```

<s:Panel width="892" height="259" title="Cadastro de Estudante" left="63" top="95">
  <mx:Form left="0" right="656" top="0" bottom="10" defaultButton="{button2}">
    <mx:FormItem label="GRR">
      <s:TextInput id="estudGrrTextInput" text="{estudante.estuGrr}"/>
    </mx:FormItem>
    <mx:FormItem label="Nome">
      <s:TextInput id="nomeTextInput" text="{estudante.nome}"/>
    </mx:FormItem>
    <mx:FormItem label="E-mail">
      <s:TextInput id="emailTextInput" text="{estudante.email}" focusOut="btn_clicl
    </mx:FormItem>
    <mx:FormItem label="Tel. Residencial">
      <s:TextInput id="telefoneResTextInput" text="{estudante.telefoneRes}" keyDown
    </mx:FormItem>
    <mx:FormItem label="Tel. Celular">
      <s:TextInput id="telefoneCelTextInput" text="{estudante.telefoneCel}" keyDown
    </mx:FormItem>
  </mx:Form>
  <mx:Form x="267" y="0" width="242" height="215" defaultButton="{button2}">
    <s:Button id="button2" width="148" label="Cadastrar" click="button2_clickHandler
    <s:Button id="buttonAlter" width="148" label="Alterar" click="buttonAlterar_clickl
    <s:Button id="buttonDel" width="148" label="Deletar" click="buttonDel_clickHandl
    <s:Button id="buttonLimpar" width="147" label="Limpar" click="buttonLimpar_clickl
  </mx:Form>
</s:Panel>

```

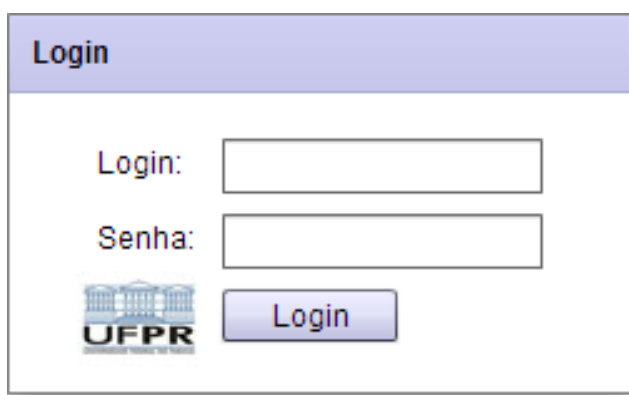
**Figura 30 - Código FLEX**

Fonte: Autor

### 3.1.6.2. Operacionalidade do Sistema

A seguir são mostradas as telas e as funcionalidades desenvolvidas para o sistema computacional.

Para o usuário poder ter acesso ao sistema deve primeiro passar pela tela de *login*, que vai verificar através de um nome e uma senha, se o usuário que está tentando entrar no sistema tem as permissões necessárias. Se o usuário tiver permissão será aberta a página inicial, caso o usuário não esteja cadastrado ou seus dados estiverem errados permanecerá na tela de *login*, figura 6.

The image shows a web-based login form. At the top, there is a purple header bar with the word "Login" in white text. Below the header, the form has a white background. It contains two input fields: the first is labeled "Login:" and the second is labeled "Senha:". To the left of the "Senha:" label is a small logo of a building with the text "UFPR" underneath it. To the right of the "Senha:" input field is a purple button with the word "Login" in white text.

**Figura 31 - Login**

Fonte: Autor

Na tela principal (figura 7) o usuário tem acesso às funcionalidades do sistema como entrar o cadastro de estudantes, organizações e estágios e fazer consulta.

O menu superior é a ferramenta que possibilita o acesso de forma rápida e intuitiva aos dados armazenados no sistema. O menu está presente em todas as telas do sistema.



**Figura 32 - Tela inicial**  
Fonte: Autor

A figura 8 apresenta a tela de cadastro de estudante, onde é realizado o gerenciamento dos dados dos estudantes que será vinculado no estágio. De forma que é obrigatório o cadastro prévio do estudante antes de cadastrá-lo no estágio. Nesta mesma tela é apresentada a lista de estudantes cadastrados.

Para o cadastro do estágio é necessário a escolha de um estudante, que estará desempenhando as atividades na organização em que estagia, por isso o cadastro de estudante é obrigatório antes do cadastro do estágio e para ter os dados de contato do estudante.

**Cadastro de Estudante**

GRR

Nome

E-mail

Tel. Residencial

Tel. Celular

**Pesquisa**   ☐ GRR ☐ Nome

GRR	Nome	E-mail	Tel. Residencial	Tel. Celular

**Figura 33 - Tela de estudante**

Fonte: Autor

A figura 9 apresenta a tela de cadastro de docente, onde é realizado o gerenciamento dos dados dos docentes do departamento de Gestão da Informação que será vinculado no estágio. De forma que é obrigatório o cadastro prévio do docente antes de cadastrá-lo no estágio. Nesta mesma tela é apresentada a lista de docentes cadastrados.

Cada estágio precisa ter um docente responsável pela COE e um docente que irá supervisionar o estágio, o cadastro de docente reúne dados que poderão ser usados para consultar quantos estágios cada docente está supervisionando e dados de contato dos docentes.

**Cadastro de Docente**

Matricula  Tel. UFPR

Nome  Tel. Residencial

E-mail  Tel. Celular

Sala  COE (S/N) ☐

**Pesquisa**   ☐ Matricula ☐ Nome

Matricula	Nome	E-mail	Sala	Tel. UFPR	Tel. Residencial	Tel. Celular	COE

**Figura 34 - Tela de docente**  
Fonte: Autor

A figura 10 apresenta a tela de cadastro de organização, onde é realizado o gerenciamento dos dados da organização que será vinculada nos estágio. De forma que é obrigatório o cadastro prévio da organização antes de cadastrá-la no estágio. Nesta mesma tela é apresentada a lista de organizações cadastradas.

O cadastro de organização é realizado para que o sistema armazene os dados necessários para a geração dos relatórios e auxílio a tomada de decisão. Os dados cadastrados serão usados para que o departamento e a coordenação entrem em contato com as organizações e saibam o perfil das organizações que contratam os estudantes do curso.

Estudante
Docente
Organização
Estagio
Configuração
Sair

### Cadastro de Organização

Organização ID  
CNPJ  
Nome  
E-mail  
Telefone  
CEP  
Logradouro

Número  
Cidade  
Bairro  
UF  
Complemento  
Tipo Organização  
Área de Atuação

Cadastrar  
Alterar  
Deletar  
Limpar

### Organização

ID	Nome	E-mail	Telefone	CNPJ	Cidade	UF

Listar todos

**Figura 35 - Tela de organização**  
Fonte: Autor

A figura 11 apresenta a tela de cadastro de tipo de organização, onde é realizado o gerenciamento dos dados dos tipos de organização que será vinculado no estágio. De forma que é obrigatório o cadastro prévio do tipo de organização antes de cadastrá-lo na organização. Nesta mesma tela é apresentada a lista de tipos de organização cadastrados.

O tipo de organização indica se a organização é privada, pública, etc.

The screenshot shows a web application interface with a top navigation bar and two main sections. The navigation bar includes links for 'Estudante', 'Docente', 'Organização', 'Estagio', 'Configuração', and 'Sair'. The first section, titled 'Cadastro de Tipo de Organização', contains a form with two input fields: 'Tipo Org. ID' and 'Tipo Organização'. To the right of these fields are four buttons: 'Cadastrar', 'Alterar', 'Deletar', and 'Limpar'. The second section, titled 'Pesquisa', contains a table with two columns: 'ID' and 'Tipo de Organização'. The table is currently empty. Below the table is a button labeled 'Listar todos'.

ID	Tipo de Organização

**Figura 36 - Tela de tipo de organização**

Fonte: Autor



A figura 12 apresenta a tela de cadastro de setor, onde é realizado o gerenciamento dos dados dos setores que será vinculado no estágio. De forma que é obrigatório o cadastro prévio do setor antes de cadastrá-lo no estágio. Nesta mesma tela é apresentada a lista de setores cadastrados.

A informação de setor, que é vinculado com o estágio se refere a qual setor da organização o estudante irá atuar durante o período de estágio. Por exemplo, informática, recursos humanos, desenvolvimento de sistemas, etc.

Estudante Docente Organização Estágio Configuração Sair

### Cadastro de Setor

Setor ID

Nome

Cadastrar

Alterar

Deletar

Limpar

### Pesquisa

Setor

Listar todos

**Figura 37 – Tela de setor**  
Fonte: Autor

A figura 13 apresenta a tela de cadastro de área de atuação, onde é realizado o gerenciamento dos dados das áreas de atuação que será vinculada na organização. De forma que é obrigatório o cadastro prévio da área de atuação antes de cadastrá-la na organização. Nesta mesma tela é apresentada a lista de áreas de atuação cadastradas.

A área de atuação foi criada para ter o controle e dados sobre que tipos de organizações estão contratando os estagiários de GI. Se as organizações são da área industrial, comércio, moda, ensino, etc.

Estudante Docente Organização Estagio Configuração Sair

**Cadastro de Área de Atuação**

Área Atuação ID

Área de Atuação

**Pesquisa**

ID	Área de Atuação

**Figura 38 - Tela de área de atuação**

Fonte: Autor

A figura 14 apresenta a tela de cadastro de agente integrador, onde é realizado o gerenciamento dos dados do agente integrador que será vinculado nos estágio. De forma que, se tiver envolvido um agente integrador no contrato, é obrigatório o cadastro prévio do agente integrador antes de cadastrá-lo no estágio. Nesta mesma tela é apresentada a lista de agentes integradores cadastrados.

O agente integrador é a organização intermediária entre o estudante e a organização contratante, ela é responsável por oferecer as vagas, convocar os alunos e realizar os contratos entre ambas as partes.

Estudante Docente Organização Estágio Configuração Sair

### Cadastro de Agente Integrador

Age. Integrador ID

Agente Integrador

### Pesquisa

ID	Agente Integrador

**Figura 39 - Tela de agente integrador**  
Fonte: Autor

A figura 15 apresenta a tela de cadastro de atividade, onde é realizado o gerenciamento dos dados das atividades que será vinculado no estágio. De forma que é obrigatório o cadastro prévio da atividade antes de cadastrá-la no estágio. Nesta mesma tela é apresentada a lista de atividades cadastradas.

As atividades serão a base para analisar as competências necessárias para o gestor da informação com relação ao mercado de trabalho e quais os conhecimentos mais requisitados pelas organizações quando resolvem contratar um gestor de informação.

**Cadastro de Atividade**

Atividade ID:

Atividade:

Descrição:

Cadastrar

Alterar

Deletar

Limpar

**Pesquisa**

ID	Atividade

Listar todos

**Figura 40 - Tela de atividade**  
Fonte: Autor

A figura 16 apresenta a tela de cadastro de estágio, onde é realizado o gerenciamento dos dados dos estágios que serão usados para controle e geração de relatórios para tomada de decisão. Nesta mesma tela é apresentada a lista de estágios cadastrados.

O cadastro de estágio reúne as informações necessárias para as análises que serão realizadas. É nesse cadastro que são vinculados os estudantes, organizações, agente integrador, setor, docente supervisor e responsável pela COE, e quais as atividades que o estudante irá executar no estágio.

**Cadastro de Estágio**

Cod ID:

Obrigatório: ☐

Estudante:

Data Inicio:

Data Fim Prevista:

Data Fim Efetiva:

Hora Inicio:

Hora Fim:

Valor Bolsa:

Doc. Supervisor:

Doc. COE:

Agente integrador:

Organização:

Setor:

Org. Supervisor:

Atividades

Selecionados

Atividades

Atividades

Pesquisa:  Busca:

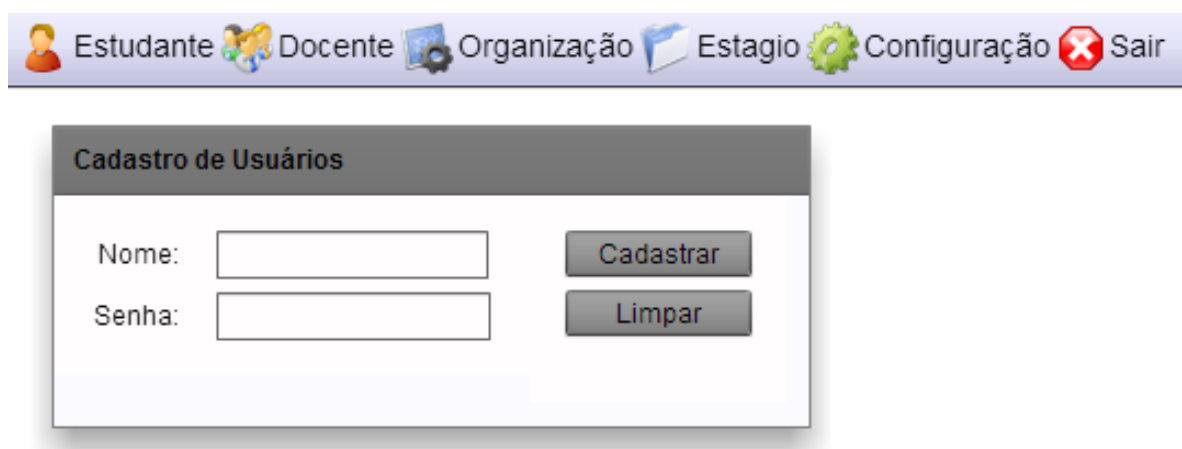
Estudante	Organização	Docente Sup	Docente COE	Data de Inicio	Data Fim Prev.	Data Fim Efet.	Valor da Bolsa

Listar todos

**Figura 41 - Tela de estágio**

Fonte: Autor

A figura 17 apresenta a tela de cadastro de usuário. O cadastro de usuário insere no banco de dados o *login* e a senha para acesso ao sistema.



A interface de usuário é composta por uma barra de navegação superior e uma janela de cadastro. A barra de navegação contém ícones e texto para: Estudante, Docente, Organização, Estagio, Configuração e Sair. A janela de cadastro, intitulada "Cadastro de Usuários", possui dois campos de entrada: "Nome:" e "Senha:". Ao lado de cada campo há um botão: "Cadastrar" para o nome e "Limpar" para a senha.

**Figura 42 - Tela de cadastro de usuário**  
Fonte: Autor

## 4. RESULTADOS E DISCUSSÃO

O resultado deste trabalho é um sistema computacional, que possibilita o gerenciamento das informações dos estágios, dos estudantes, das organizações e quais as atividades realizadas nos estágios.

Com a utilização de um ambiente *web*, com desenvolvimento de páginas em linguagem *Java*, *Flex* e o banco de dados *MySql*, tornou o sistema acessível de qualquer lugar que tenha *internet*.

O sistema auxilia no levantamento de quais as habilidades estão sendo requisitadas pelo mercado de trabalho e quais as organizações mais precisam dos gestores de informação, podendo dessa forma direcionar os estudos e matérias para formar melhor os estudantes de GI.

## 5. CONCLUSÃO

A gestão da informação nas organizações é fundamental para que se alcancem vantagens competitivas, informações sobre o mercado em que atua, subsidio para a tomada de decisão e gerenciamento dos dados que são pertinentes aos processos da organização.

Nesse contexto, é incontestável o papel da tecnologia de informação e comunicação como suporte para as atividades da gestão da informação e do conhecimento. As novas tecnologias facilitam os processos de coleta, disseminação e tratamento de dados de uma organização.

Mas para que essas tecnologias sejam usadas de forma com que apresentem subsídio a tomada de decisão, desenvolvimento de estratégias e gerem vantagens competitivas, é necessária uma análise dos processos da organização para saber como as tecnologias podem dar suporte as atividades, pois um bom sistema de informação é aquele que auxilia nos processos da organização, centralizando as informações e deixando-as acessíveis a quem é de direito, tornando o processo mais ágil e seguro.

Analisando os problemas levantados ficou claro a necessidade de desenvolver um sistema informatizado para gerenciar as informações sobre os estágios, estudantes e organizações. Para subsidiar a tomada de decisão e o controle dessas informações. Sendo proposto construir uma ferramenta responsável pelo gerenciamento das informações dos estágios, estruturando os dados e visando à extração de informações para apoiar a tomada de decisão.

O objetivo geral, que era desenvolver um sistema de informação computacional para gerenciar as informações dos estágios realizados pelos estudantes do curso de Gestão da Informação da UFPR foi atingido.

O objetivo específico que visava controlar de forma sistematizada e informatizada os estágios foi alcançado, já que através do sistema desenvolvido, é possível gerenciar as informações dos estágios.

O objetivo específico que abordava o auxílio à tomada de decisão para reforma do currículo será avaliada no momento em que os docentes forem revisar a grade curricular do curso com o apoio do sistema computacional.



O objetivo específico referente aos relatórios que tinha sido proposto foi desenvolvido e está acessível ao usuário. Porém com o uso do sistema provavelmente serão identificadas novas necessidades e será necessário desenvolver novos relatórios para atender os usuários.

Por ser um sistema web, o mesmo fica disponível a qualquer momento para o acesso de qualquer lugar, que tenha *internet*, sendo possível realizar consultas, editar dados cadastrais, deixando-os sempre atualizados e avaliar quais atividades está sendo desenvolvidas nos estágios. Sendo um diferencial para o curso, podendo analisar tendências e quais os principais conhecimentos as organizações esperam de um Gestor da Informação.

Para a especificação do sistema e modelagem do banco de dados, foram usados métodos consolidados no mercado, como o diagrama entidade-relacionamento, o dicionário de dados e os casos de uso. O que facilitou a modelagem do sistema, a interação com os usuários e irá facilitar o entendimento do sistema para futuros desenvolvedores que forem dar continuidade no projeto.

Este trabalho ajudou a aplicar os conhecimentos adquiridos em sala, o que possibilitou concretizar o aprendizado com uma aplicação prática. E engrandecer os conhecimentos pessoais, sobre gestão da informação, sistemas de informação, termos técnicos, linguagem de programação, análise de requisitos e necessidades e interação com usuários. Conhecimentos que são importantes para o desenvolvimento de um sistema.

Por fim, o desenvolvimento e implantação desse sistema contribuem para a gestão das informações dos estágios do curso de GI. Possibilitando, as pessoas que tem acesso, a realização de consultas sobre as atividades realizadas e a situação dos estágios, informações dos estudantes, docentes e organizações.

## REFERENCIAS

ABRAÃO, J. I.; PINHO, D. L. M. **As transformações do trabalho e desafios teórico metodológicos da Ergonomia**. Estudos de Psicologia, v. 07 (Número Especial), p. 45-52, 2002.

ALBERTIN, Alberto Luiz; ALBERTIN, Rosa Maria de Moura. **Benefícios do uso de tecnologia de informação para o desempenho empresarial**. *Rev. Adm. Pública* . 2008.

ALVES, Robson de Paula; FALSARELLA, Orandi Mina **Modelo conceitual de inteligência organizacional aplicada à função manutenção**. *Gest. Prod.*, Jun 2009, vol.16, no.2, p.313-324.

AUDY, Jorge; LEDERER, Albert; BRATCHER, Annaliese. **Princípios da Aprendizagem Organizacional aplicados ao Planejamento de Sistemas de Informação**. Disponível em:  
<<http://www.serprofessoruniversitario.pro.br/m%C3%B3dulos/aprendizagemorganizacional/princ%C3%ADpios-da-aprendizagem-organizacional-aplicados-ao-planejame>> Acesso em: 30 de maio de 2012

BARRANCO DE AREBA, Jesús **Metodología Del Análisis Estructurado de Sistemas**. 2ª ed. Madrid: Universidad de Comillas, 2002.

BATISTA E. O. **Sistemas de informação**. São Paulo: Saraiva, 2004.

BRAGA, Ascensão. **A Gestão da Informação**. Millenium. N.º 19, junho de 2000. Disponível em:  
<<http://repositorio.ipv.pt/bitstream/10400.19/903/1/A%20GEST%C3%83O%20DA%20INFORMA%C3%87%C3%83O.pdf>> Acesso em 25 de maio de 2012.

CARVALHO, G. M. R.; TAVARES, M. S. **Informação & conhecimento**: uma abordagem organizacional. Rio de Janeiro: Qualitymark, 2001.

COSTA, Rogério Luís de C. **SQL**: Guia Prático. 2ª ed. Rio de Janeiro: Brasport, 2006.

DATE, C. J. **Introdução a sistemas de bancos de dados**. 8. ed. Rio de Janeiro: Campus, 2003.

DAVENPORT, E.; CRONIN, B. **Knowledge management: semantic drift or conceptual shift?** Journal of Education for Library and Information Science, v. 41, n.4, p. 294-306, Fall 2000. Disponível em: <<http://www.jstor.org/stable/40324047>> Acesso em: 04 de abril de 2012

DIAS, M. P.; CARVALHO, F. J. **A visualização da informação e a sua contribuição para a ciência da informação**. Ciência da Informação, Brasília, v.8, n. 5. Brasil, 2007.

DRUCKER, P. **Gerindo para o Futuro**. Difusão Cultural. Lisboa, 1993.

FARIAS, Anderson Rodrigo. **História da Oracle**. Disponível em <<http://www.devmedia.com.br/historia-da-oracle/4685>> Acesso em: 05 de outubro de 2012.

GUEDES, Gilleanes T. A. **UML**: uma abordagem prática. 2. ed. São Paulo: Novatec, 2006.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 5. ed. Porto Alegre: Sagra Luzzatto, 2004.

HORSTMANN, Cay S. **Big Java**. Porto Alegre: Bookman, 2004.

KORTH, Henry F; SILBERSCHATZ, Abraham. **Sistema de banco de dados**. São Paulo: McGraw-Hill, c1989. xv, 582.

LAUDON, K. C., 1944-. **Sistemas de informação gerenciais**. 7. ed. São Paulo: Prentice Hall, 2007.

LAUDON, K.C., LAUDON, J. P. **Essentials of Management Information Systems: transforming business and management**. 3. ed., Upper Saddle River, NJ: Prentice-Hall, 1999.

LEHMKUHL, G. T.; VEIGA C. R.; RADOGL. J. V. **O Papel da tecnologia da informação como auxílio à engenharia e gestão do conhecimento**. *Revista Brasileira de Biblioteconomia e Documentação*, Nova Série, São Paulo, v.4, n.1, p. 59-67, jan./jun. 2008.

MAFFEO, Bruno. **Engenharia de software e especificação de sistemas**. Rio de Janeiro: Campus, 1992.

MARCHIORI, P. Z. . **A ciência e a gestão da informação: compatibilidades no espaço profissional**. *Ciência da Informação*, Brasília, v. 31, n. 2, p. 72-79, 2002.

MENEZES, Eduardo Diatahy Bezerra de. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2002.

MONTEIRO, Nabor. A.; FALSARELLA, O. M. **Um modelo de gestão da informação para aprendizagem organizacional em projetos empresariais**. *Perspectivas em Ciência da Informação*, v. 12, n.2, p. 81-97, maio/ago. 2007.

MONTMOLLIN, M. **L'Ergonomie**. Paris: La Découverte, 1990.

MORAES, A. **Quando a primeira sociedade de ergonomia faz 50 anos, a IEA chega aos 40, a Associação Brasileira de Ergonomia debuta com 16.** Disponível em: <<http://labeledi.vilabol.uol.com.br/Fhistorico.html>> Acesso em: 12 de dezembro de 2011.

MORESI, Eduardo Amadeu Dutra. **Inteligência organizacional:** um referencial integrado. *Ci. Inf.*, Ago 2001, vol.30, no.2, p.35-46.

NIEDERAUER, Juliano; PRATES, Rubens. **MYSQL 5** - Guia de consulta rápida. São Paulo: Novatec, 2006.

NONAKA I, TAKEUCHI H. **Criação de conhecimento na empresa.** Rio de Janeiro: Campus; 2000.

OLIVEIRA, Roberto de Jesus. **Vamos mudar a cultura das empresas!** Disponível em: <<http://imasters.com.br/artigo/11836/e-commerce/vamos-mudar-a-cultura-das-empresas>> Acesso em: 04 de abril de 2012

O'BRIEN, James A., 1936-; MOREIRA, Célio Knipel. **Sistemas de informação:** e as decisões gerenciais na era da internet. 2. ed. São Paulo: Saraiva, 2004.

PADOVEZE, Clóvis Luís. **Sistemas de informações contábeis:** fundamentos e análise. 3ª ed. São Paulo: Atlas, 2002.

PAULA FILHO, Wilson de Pádua. **Engenharia de software:** fundamentos, métodos e padrões. 2. ed. Rio de Janeiro: LTC, 2003.

PIETRO, G. A., **Utilização de padrões de projeto na reengenharia de sistemas.** Universidade Federal de São Carlos, 2001.

PRESSMAN, Roger S.. **Engenharia de software.** 6ª ed. São Paulo: McGraw Hill, 2006.

PRÓ-REITORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL.  
Coordenação-Geral de Estágios. **Lei de Estágio**. Disponível em:  
<[http://www.estagios.ufpr.br/documentos/Lei\\_N\\_11788.doc](http://www.estagios.ufpr.br/documentos/Lei_N_11788.doc)>. Acesso em: 20 de  
abril de 2012.

PRÓ-REITORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL.  
Coordenação-Geral de Estágios. **Manual de Estágio**. Disponível em:  
<<http://www.estagios.ufpr.br/Cadernos%20Atividades%20Formativas/manualestagiosfinal.pdf>>. Acesso em: 20 de abril de 2012.

PRÓ-REITORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL.  
Coordenação-Geral de Estágios. **Normatização**. Disponível em: <  
<http://www.estagios.ufpr.br/normatizacao.htm>>. Acesso em: 20 de abril de 2012.

PRÓ-REITORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL.  
Coordenação-Geral de Estágios. **Procedimentos**. Disponível em:  
<<http://www.estagios.ufpr.br/procedimentos.htm>> Acesso em: 20 de abril de 2012.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**.  
3 ed. Rio de Janeiro: Brasport, 2005.

REZENDE, Denis Alcides. **Planejamento de sistemas de informação e informática**: guia prático para planejar a tecnologia da informação integrada ao planejamento estratégico das organizações. 3. ed. São Paulo: Atlas, 2008.

ROBALO SANTOS, António J. **Gestão Estratégica**: Conceitos, modelos e instrumentos. Lisboa: Escolar, 2008.

SETZER, V.W. **Dado, Informação, Conhecimento e Competência**. Rio de Janeiro: DataGramZero, nº.zero, dez. 1999.

SAID, Ricardo **Curso de lógica de programação**. São Paulo: Digerati Books: 2007.

SILVA, C. G. (2007) **Considerações sobre o uso de visualização de informação no auxílio à gestão de informação**. XXXIV SEMISH - Seminário Integrado de Software e Hardware. Rio de Janeiro - RJ, 2007.

TURBAN Efraim; SHARDA Ramesh; ARONSON Jay E.; KING David **Business Intelligence**: Um enfoque gerencial para a inteligência do negócio. Porto Alegre:Bookman, 2008.

TURBAN, Efraim; MCLEAN, Ephraim; WETHERBER, James **Tecnologia da informação para gestão**: transformando os negócios na economia digital. 6. ed. Porto Alegre: Bookman, 2010.

TURBAN, Efraim; RAINER JÚNIOR, R. Kelly; POTTER, Richard E. **Introdução a Sistemas de Informação**: uma abordagem gerencial. Rio de Janeiro: Elsevier/Campus, 2007.

VIRGILIO, Rodrigo. **Sistemas de Informação** - Centro Universitário Claretiano. Disponível em <<http://www.slideshare.net/guest043b5a/adobe-flex-presentation>> Acesso em: 05 de outubro de 2012.

WILLIAM, David. **A História Do HTML** Disponível em: <<http://www.frontendbrasil.com.br/artigos/a-historia-do-html/>> Acesso em: 03 de outubro de 2012.

## **APÊNDICE A – CÓDIGO DO SISTEMA PARA “ESTUDANTE”**



Abaixo são apresentados os códigos em JAVA e FLEX usado para manipular os dados do estudante.

### **Classe JAVA “Estudante”**

```
package br.com.horochovec.model;
import java.util.List;
public class Estudante {
    private String estuGrr;
    private String estuGrrNew;
    private String nome;
    private String email;
    private String telefoneRes;
    private String telefoneCel;
    private List<Estagio> listaEstagio;
    public String getEstuGrrNew() {
        return estuGrrNew;
    }
    public void setEstuGrrNew(String estuGrrNew) {
        this.estuGrrNew = estuGrrNew;
    }
    public String getEstuGrr() {
        return estuGrr;
    }
    public void setEstuGrr(String estuGrr) {
        this.estuGrr = estuGrr;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```

```

    }
    public String getTelefoneRes() {
        return telefoneRes;
    }
    public void setTelefoneRes(String telefoneRes) {
        this.telefoneRes = telefoneRes;
    }
    public String getTelefoneCel() {
        return telefoneCel;
    }
    public void setTelefoneCel(String telefoneCel) {
        this.telefoneCel = telefoneCel;
    }
    public List<Estagio> getListaEstagio() {
        return listaEstagio;
    }
    public void setListaEstagio(List<Estagio> listaEstagio) {
        this.listaEstagio = listaEstagio;
    }
}

```

### **Classe JAVA “EstudanteService”**

```

package br.com.horochovec.service;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.StringTokenizer;
import br.com.horochovec.dao.Dao;
import br.com.horochovec.model.Estudante;
import br.com.horochovec.model.Organizacao;
import br.com.horochovec.model.Professor;
public class EstudanteService {
    private static ArrayList listEstudante = new ArrayList();
    /**
     * Adiciona um contato a lista
     * @param estudante
     */
    public void addEstudante(Estudante estudante) {

```

```

        boolean grrExiste = false;
        String estuGrr = UtilCoe.TestaNullInt(estudante.getEstuGrr());
        String nome = UtilCoe.TestaNullString(estudante.getNome());
        String email = UtilCoe.TestaNullString(estudante.getEmail());
        String telefoneRes =
UtilCoe.TestaNullString(estudante.getTelefoneRes());
        String telefoneCel =
UtilCoe.TestaNullString(estudante.getTelefoneCel());
        String query = "SELECT EST_GRR FROM estudante";
        Dao dao = new Dao();
        try{
            ResultSet rst = dao.readSql(query);
            while(rst.next()){
                if(estuGrr == rst.getString("EST_GRR") ||
estuGrr.equals(rst.getString("EST_GRR"))){
                    grrExiste = true;
                }
            }
        }catch (Exception e) {
            System.out.println("Erro Execução Dao 'Estudante'");
        }
        query = "";
        if(grrExiste == false){
            query = "INSERT INTO `coe`.`estudante` (`EST_GRR`,
`EST_Nome`, `EST_Email`, `EST_TelefoneRes`, `EST_Celular`) " +
VALUES (" + estuGrr + ", " + nome + ", " + email + ", " + telefoneRes + ", " +
telefoneCel + ");";
            System.out.println(query);
        }
        try{
            dao.executaSql(query);
            System.out.println("Executou Dao 'Estudante'");
        }catch (Exception e) {
            System.out.println("Erro Execução Dao 'Estudante'");
        }
        listEstudante.add(estudante);
    }
    /**
    * Deleta um contato a lista

```

```

* @param organizacao
*/
public void delEstudante(Estudante estudante) {
    String id = "";
    String estudantelDs = estudante.getEstuGrr();
    String query = "";
    Dao dao = new Dao();
    StringTokenizer st = new StringTokenizer(estudantelDs, ",");
    while (st.hasMoreTokens()){
        id = st.nextToken();
        System.out.println("ID do array: " + id);
        query = "DELETE FROM `coe`.`estudante` WHERE
`EST_GRR`='" + id + "','";
        try{
            dao.executaSql(query);
            System.out.println("Executou Dao 'Del Estudante");
        }catch (Exception e) {
            System.out.println("Erro Execução Dao 'Del
Estudante");
        }
    }
}
/**
* Altera um contato a lista
* @param estudante
*/
public void alteraEstudante(Estudante estudante) {
    boolean grrExiste = false;
    String estuGrr = UtilCoe.TestaNullString(estudante.getEstuGrr());
    String nome = UtilCoe.TestaNullString(estudante.getNome());
    String email = UtilCoe.TestaNullString(estudante.getEmail());
    String telefoneRes =
UtilCoe.TestaNullString(estudante.getTelefoneRes());
    String telefoneCel =
UtilCoe.TestaNullString(estudante.getTelefoneCel());
    Dao dao = new Dao();
    String query = "";

    if(grrExiste == false){

```

```

        query = "UPDATE `coe`.`estudante` SET\n" +
            "`EST_GRR`=" +
UtilCoe.TestaNullString(estudante.getEstuGrrNew()) + ",\n" +
            "`EST_Nome`=" + nome + ",\n" +
            "`EST_Email`=" + email + ",\n" +
            "`EST_TelefoneRes`=" + telefoneRes + ",\n" +
            "`EST_Celular`=" + telefoneCel + "\n" +
            "WHERE `EST_GRR`=" + estuGrr + ";";

        System.out.println(query);
        dao.executaSql(query);
    }
    try{
        dao.executaSql(query);
        System.out.println("Executou Dao 'Altera Estudante'");
    }catch (Exception e) {
        System.out.println("Erro Execução Dao 'Altera Estudante'");
    }
    listEstudante.add(estudante);
}
/**
 * Retorna a lista de Estudantes
 * @return
 */
public ArrayList getListEstudante() {
    listEstudante = carregaEstudante();
    return listEstudante;
}
public ArrayList carregaEstudante(){
    ArrayList listEstudante = new ArrayList();
    String query = "SELECT * FROM estudante ORDER BY
UCase(EST_Nome)";
    Dao dao = new Dao();
    try{
        ResultSet rst = dao.readSql(query);
        while(rst.next()){
            Estudante estudante = new Estudante();
            estudante.setEstuGrr(rst.getString("EST_GRR"));
            estudante.setNome(rst.getString("EST_Nome"));
            estudante.setEmail(rst.getString("EST_Email"));

```

```

        estudante.setTelefoneRes(rst.getString("EST_TelefoneRes"));
        estudante.setTelefoneCel(rst.getString("EST_Celular"));
        listEstudante.add(estudante);
    }
} catch (Exception e) {
    System.out.println("Erro Execução Dao 'Estudante'");
}
return listEstudante;
}
/**
 * Retorna a lista de contatos Filtrados
 * @return
 */
public ArrayList getListEstudanteFiltrado(String chek, String valorConsulta)
{
    listEstudante = carregaEstudanteFiltrado(chek, valorConsulta);
    return listEstudante;
}
public ArrayList carregaEstudanteFiltrado(String chek, String
valorConsulta){
    ArrayList listEstudante = new ArrayList();
    String query = "";
    if(chek.equals("true")){
        query = "SELECT * from estudante WHERE EST_Nome LIKE
'" + valorConsulta + "%' ORDER BY UCase(EST_Nome)";
        Dao dao = new Dao();
        try{
            ResultSet rst = dao.readSql(query);
            while(rst.next()){
                Estudante estudante = new Estudante();
                estudante.setEstuGrr(rst.getString("EST_GRR"));
                estudante.setNome(rst.getString("EST_Nome"));
                estudante.setEmail(rst.getString("EST_Email"));
                estudante.setTelefoneRes(rst.getString("EST_TelefoneRes"));
                estudante.setTelefoneCel(rst.getString("EST_Celular"));
                listEstudante.add(estudante);
            }
        } catch (Exception e) {

```

```

        System.out.println("Erro Execução Dao 'Pesquisa
Estudante");
    }
    }else{
        query = "SELECT * from estudante WHERE EST_GRR LIKE
'" + valorConsulta + "' ORDER BY UCCase(EST_Nome)";
        Dao dao = new Dao();
        try{
            ResultSet rst = dao.readSql(query);
            while(rst.next()){
                Estudante estudante = new Estudante();
                estudante.setEstuGrr(rst.getString("EST_GRR"));
                estudante.setNome(rst.getString("EST_Nome"));
                estudante.setEmail(rst.getString("EST_Email"));
                estudante.setTelefoneRes(rst.getString("EST_TelefoneRes"));
                estudante.setTelefoneCel(rst.getString("EST_Celular"));
                listEstudante.add(estudante);
            }
        }catch (Exception e) {
            System.out.println("Erro Execução Dao 'Pesquisa
Estudante");
        }
    }
    return listEstudante;
}
}

```

### Classe FLEX “EstudanteComp”

```

<?xml version="1.0" encoding="utf-8"?>
<s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="1372"
height="576" xmlns:services="services.*" xmlns:valueObjects="valueObjects.*"
    initialize="inicio()">
    <fx:Script>
        <![CDATA[
            import mx.collections.ArrayCollection;
            import mx.controls.Alert;

```

```

import mx.events.ListEvent;
import mx.events.ValidationResultEvent;
import mx.rpc.events.ResultEvent;
import valueObjects.Estudante;
private var grrOld:String;
protected function
button_clickHandler(event:MouseEvent):void
{
    getListEstudResult.token =
EstudService.getListEstudante();
}
protected function
button2_clickHandler(event:MouseEvent):void{
    var estudante : valueObjects.Estudante = new
valueObjects.Estudante();
    if(estudGrrTextInput.text.length == 11){
        if(nomeTextInput.text == null || nomeTextInput.text.length < 3){
            Alert.show("O campo Nome é obrigatório!");
            return;
        }
        estudante.estuGrr = estudGrrTextInput.text;
        estudante.nome = nomeTextInput.text;
        estudante.email = emailTextInput.text;
        estudante.telefoneRes = telefoneResTextInput.text;
        estudante.telefoneCel = telefoneCelTextInput.text;
        addEstudResult.token =
EstudService.addEstudante(estudante);
    }else{
        Alert.show("GRR deve conter 11 caracteres!");
        return;
    }
}
protected function
buttonAlterar_clickHandler(event:MouseEvent):void
{
    var estudante : valueObjects.Estudante = new
valueObjects.Estudante();
    if(estudGrrTextInput.text.length == 11){
        if(nomeTextInput.text == null ||
nomeTextInput.text.length < 3){

```



```

        Alert.show("O campo Nome é obrigatório!");
        return;
    }
    estudante.estuGrr = grrOld;
    estudante.estuGrrNew = estudGrrTextInput.text;
    estudante.nome = nomeTextInput.text;
    estudante.email = emailTextInput.text;
    estudante.telefoneRes = telefoneResTextInput.text;
    estudante.telefoneCel = telefoneCelTextInput.text;
    addEstudResult.token =
EstudService.alteraEstudante(estudante);
    }else{
        Alert.show("GRR deve conter 11 caracteres!");
        return;
    }
    }
    protected function
buttonLimpar_clickHandler(event:MouseEvent):void
    {
        var estudante : valueObjects.Estudante = new
valueObjects.Estudante();
        estudante.estuGrr = null;
        estudante.nome = null;
        estudante.email = null;
        estudante.telefoneRes = null;
        estudante.telefoneCel = null;
        addEstudResult.token =
EstudService.alteraEstudante(estudante);
    }
    protected function
buttonDel_clickHandler(event:MouseEvent):void
    {
        var estudante : valueObjects.Estudante = new
valueObjects.Estudante();
        var array:Array = grid.selectedItems;
        var estudanteIDs:String = "";
        for (var i:Number = 0; i < grid.selectedItems.length;
i++){
            estudanteIDs = estudanteIDs +
array[i].estuGrr.toString() + ",";

```

```

    }
    estudante.estuGrr = estudanteIds.substr(0,
estudanteIds.length -1);
    addEstudResult.token =
EstudService.delEstudante(estudante);
    }
    protected function
addEstudResult_resultHandler(event:ResultEvent):void
    {
        estudante = new valueObjects.Estudante();
        getListEstudResult.token =
EstudService.getListEstudante();
    }
    protected function inicio():void{
        getListEstudResult.token =
EstudService.getListEstudante();
    }
    protected function mudaGrid(event:ListEvent):void{
        grrOld = estudante.estuGrr;
    }
    protected function carregaPesquisa(event:MouseEvent):void{
        getListEstudResult.token =
EstudService.getListEstudanteFiltrado(chekNome.selected.toString(),
inputPesquisa.text);
    }
    protected function
chekNome_changeHandler(event:Event):void{
        if(chekNome.selected == true){
            chekGrr.selected = false;
        }else{
            chekGrr.selected = true;
        }
    }
    protected function chekGrr_changeHandler(event:Event):void{
        if(chekGrr.selected == true){
            chekNome.selected = false;
        }else{
            chekNome.selected = false;
        }
    }
}

```

```

//+++++++-----VALIDAÇÕES-----+++++++
public var teclas:int;
public function myKeyDownHandler( event:KeyboardEvent
):void {
    if(event.keyCode != 8){
        if(telefoneResTextInput.text.length == 4){
            telefoneResTextInput.insertText("-");
        }
    }else{
        //Alert.show("BackSpace?");
    }
}
public var teclasC:int;
public function myKeyDownHandleC( event:KeyboardEvent
):void {
    if(event.keyCode != 8){
        if(telefoneCelTextInput.text.length == 4){
            telefoneCelTextInput.insertText("-");
        }
    }else{
        //Alert.show("BackSpace?");
    }
}
private function
emailValidator_valid(evt:ValidationResultEvent):void {
    emailTextInput.errorString = "";
}
private function
emailValidator_invalid(evt:ValidationResultEvent):void {
    emailTextInput.errorString = evt.message;
}
private function btn_click():void {
}

//+++++++-----VALIDAÇÕES-----+++++++
[Bindable]
public var TesteArray:ArrayCollection = new ArrayCollection();
protected function
boPesquisa0_clickHandler(event:MouseEvent):void {
    var contaTeste:int = 0;

```

```

                                for each (var value:String in
getListEstudResult.lastResult){

        TesteArray.addItem(getListEstudResult.lastResult[contaTeste].nome.toString());

                                Alert.show(TesteArray[contaTeste]);
                                contaTeste++;
                                }
        }

    ]]>
</fx:Script>
<fx:Declarations>
    <s:CallResponder id="getListEstudResult"/>
    <services:EstudanteService id="EstudService"
fault="Alert.show(event.fault.faultString + '\n' + event.fault.faultDetail)"
showBusyCursor="true"/>
    <valueObjects:Estudante id="estudante"/>
    <s:CallResponder id="addEstudResult"
result="addEstudResult_resultHandler(event)"/>
    <mx:EmailValidator id="emailValidator"
valid="emailValidator_valid(event);" invalid="emailValidator_invalid(event);" />
</fx:Declarations>
<fx:Binding source="grid.selectedItem as Estudante"
destination="estudante"/>
<s:Panel width="892" height="259" title="Cadastro de Estudante" left="63"
top="95">
    <mx:Form left="0" right="656" top="0" bottom="10"
defaultButton="{button2}">
        <mx:FormItem label="GRR">
            <s:TextInput id="estudGrrTextInput"
text="{estudante.estuGrr}" maxChars="11"/>
        </mx:FormItem>
        <mx:FormItem label="Nome">
            <s:TextInput id="nomeTextInput"
text="{estudante.nome}"/>
        </mx:FormItem>
        <mx:FormItem label="E-mail">
            <s:TextInput id="emailTextInput"
text="{estudante.email}" focusOut="btn_click()"/>
        </mx:FormItem>

```

```

        <mx:FormItem label="Tel. Residencial">
            <s:TextInput id="telefoneResTextInput"
text="{estudante.telefoneRes}" keyDown="myKeyDownHandler(event)" restrict="0-9" maxChars="9"/>
        </mx:FormItem>
        <mx:FormItem label="Tel. Celular">
            <s:TextInput id="telefoneCelTextInput"
text="{estudante.telefoneCel}" keyDown="myKeyDownHandleC(event)" restrict="0-9" maxChars="9"/>
        </mx:FormItem>
    </mx:Form>
    <mx:Form x="267" y="0" width="242" height="215"
defaultButton="{button2}">
        <s:Button id="button2" width="148" label="Cadastrar"
click="button2_clickHandler(event)"/>
        <s:Button id="buttonAlter" width="148" label="Alterar"
click="buttonAlterar_clickHandler(event)"/>
        <s:Button id="buttonDel" width="148" label="Deletar"
click="buttonDel_clickHandler(event)"/>
        <s:Button id="buttonLimpar" width="147" label="Limpar"
click="buttonLimpar_clickHandler(event)"/>
    </mx:Form>
</s:Panel>

<s:Panel y="362" left="63" width="892" height="277" title="Pesquisa">
    <mx:DataGrid id="grid" left="5" right="5" top="5" bottom="30"
dataProvider="{getListEstudResult.lastResult}"
allowMultipleSelection="true"
change="mudaGrid(event)">
        <mx:columns>
            <mx:DataGridColumn width="60" headerText="GRR"
dataField="estuGrr"/>
            <mx:DataGridColumn width="100" headerText="Nome"
dataField="nome"/>
            <mx:DataGridColumn width="180" headerText="E-mail"
dataField="email"/>
            <mx:DataGridColumn width="50" headerText="Tel.
Residencial" dataField="telefoneRes"/>
            <mx:DataGridColumn width="50" headerText="Tel.
Celular" dataField="telefoneCel"/>
        </mx:columns>
    </mx:DataGrid>

```

```

        <s:Button x="-1" y="163" label="Listar todos" left="5" bottom="5"
id="button" click="button_clickHandler(event)"/>
        <s:Button id="boPesquisa" x="340" y="-26" label="Busca"
click="carregaPesquisa(event)"/>
        <s:TextInput id="inputPesquisa" x="80" y="-27" width="252"/>
        <s:CheckBox id="chekNome" x="429" y="-18" label="Nome"
selected="true" change="chekNome_changeHandler(event)"/>
    </s:Panel>
    <s:CheckBox id="chekGrr" x="493" y="361" label="GRR"
change="chekGrr_changeHandler(event)"/>
</s:Group>

```